DETECTING PARKINSONS DISEASE USING MACHINE LEARNING

(TEAM ID: PNT2022TMID34835)

PROJECT REPORT

Submitted by

FATHIMA SAJA S. (962819104039)

ANISHA SHERLIN S R. (962819104015)

AJITHA MARY V. (962819104008)

DHARSHINI E A. (962819104026)

in partial fulfillment for the award of degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING, NAGERCOIL

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2022

INDEX

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING**
   - 7.1 Feature
   - 7.2 Code
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

CHAPTER-1

INTRODUCTION

Parkison’s disease (PD) is the second most common disease after Alzheimer’s and it is anticipated that the prevalence of PD is going to increase due to population ageing. The loss of dopaminergic neurons can reach up to 50% at the time of clinical diagnosis and rapidly increases completing by 4 years post-diagnosis. Any neuroprotective strategies that may emerge in the near future could be too late to effectively slow down the neurodegenerative process. Therefore, early objective diagnostic markers are critically needed. Amongst many other symptoms, PD manifests itself through speech disorders, which can be observed as early as 5 years before the diagnosis. Investigations show that Parkinsonian vocal dysfunction can be characterized by: reduced vocal tract volume and reduced tongue flexibility, significantly narrower pitch range, longer pauses and smaller variations in pitch range, voice intensity level, and articulation rate. Therefore, automated acoustic analysis is considered by many researchers as an important non-invasive tool for PD screening. To this end, acoustic analysis aims at solving either regression or classification task: PD severity evaluation based on vocal function assessment from audio samples, as in the Interspeech 2015 computational paralinguistics challenge, or early detection of PD by learning to classify audio samples into healthy control (HC) or PD cases.

1.1 Project Overview

The Parkinson's disease is progressive neuro degenerative disorder that affects a lot only people significantly affecting their quality of life. It mostly affect the motor functions of human. The main motor symptoms are called "parkinsonism" or "parkinsonian syndrome". The symptoms of Parkinson's disease will occur slowly, the symptoms include shaking, rigidity, slowness of movement and difficulty with walking, Thinking and behavior change, Depression and anxiety are also common. There is a model for detecting Parkinson's using voice. The deflections in the voice will confirm the symptoms of Parkinson's disease. In our model, a huge amount of data is collected from the normal person and also previously affected person by Parkinson's disease, these data is trained using machine learning algorithms. From the whole data 80% is used for training and 20% is used for testing. The data of any person can be entered in db to check whether the person is affected by Parkinson’s disease or not. There are 24 columns in the data set each column will indicate the symptom values of a patient except the status column. The status column has 0's and 1's.those values will decide the person is effected with Parkinson’s disease. 1's indicate person is effected, 0's indicate normal conditions.

1.2 Purpose

The purpose of the project is used to detect Parkinson disease using voice based detection by using the value extracted in voice to detect Parkinson disease by the patients from home.

CHAPTER 2

LITERATURE SURVEY

1. Parkinson disease is a neural disease. It prompts shaking of the hands, difficulty to walk, balance with coordination. No medical treatment is available in the high-level stage. X-ray, CT scan and blood tests report are not sufficiently results available in the early stage. About two trillion community are alive in Parkinson's disease (PD) in the U.K., which is the highest number of people affected are pinpointed to have different sclerosis, solid dystrophy and Lou Gehrig's illness. This is relied upon to ascend to 1.5 million by 2040. Around the 75,000 Americans are diagnosis PD with every year. It is very important to predict Parkinson's disease early so that important treatment can be done. The purpose of work is to detect Parkinson disease, where we aimed to identify disease in early prediction using clinical imaging that incorporate the use of Machine learning techniques. A comparative analysis done with various Machine Learning classifier algorithms like XG Boost, Random Forest, KNN, SVM are the best model is proposed which is used to make predictions and find accuracy. We are observed that Random Forest provides better performance with an accuracy of 90%. Automatic detection with more accuracy will make screening for Parkinson disease as cost effective and efficient manner facilitates to use appropriate and fast solutions.

2. Parkinson's disease (PD) has affected millions of people worldwide and is more prevalent in people, over the age of 50. Even today, with many technologies and advancements, early detection of this disease remains a challenge. This necessitates a need for the machine learning-based automatic approaches that help clinicians to detect this disease accurately in its early stage. Thus, the focus of the research paper is to provide an insightful survey and compare the existing computational intelligence techniques used for PD detection. To save time and increase treatment efficiency, classification has found its place in PD detection. The existing knowledge review indicates that many classification algorithms have been used to achieve better results, but the problem is to identify the most efficient classifier for PD detection. The challenge in identifying the most appropriate classification algorithm lies in their application on local dataset. Thus, in the paper three types of classifiers, namely, Multilayer Perceptron, Support Vector Machine and K-nearest neighbor have been discussed on the benchmark (voice) dataset to compare and to know which of these classifiers is the most efficient and accurate for PD classification. The Voice input dataset for these classifiers has been obtained from UCI machine learning repository. ANN with Levenberg–Marquardt algorithm was found to be the best classifier, having highest classification accuracy (95.89%).

3. Parkinson's disease is a degenerative disease that leads to brain disorder and nonfunctioning of different body parts. Deep learning tools like artificial neural network (ANN), convolution neural network (CNN), regression Analysis (RA), and so on, has been considered to a great extent in recent days. Several data sets based on the motor and nonmotor symptoms are applied to different classifier for correct identification of Parkinson's patient from healthy people. In the paper, hybridization of two deep learning tools such as, RA and ANN are done for effective diagnosis of the disease by probability estimation. The communal merits of individual approaches of the existing approaches are realized in this context for accurate probability estimation. Data preprocessing and probability estimation of preprocessed data is done in RA. The second existing approach is used to identify the PD patient by comparing with a predefined threshold value of a neuron. The estimation is performed

on the data set of speech recognition, iron content, and pulse rate among a group of people. The proposed approach is compared with the existing approaches like, SVM and k-NN classifier. The computed result reveals the superiority of the proposed algorithm with 93.46% accuracy.

4. Nowadays, an important research effort in healthcare biometrics is finding accurate biomarkers that allow developing medical-decision support tools. These tools help to detect and supervise illnesses like Parkinson's disease (PD). This paper contributes to this effort by analyzing a convolution neural network (CNN) for PD detection from drawing movements. This CNN includes two parts: feature extraction (convolution layers) and classification (fully connected layers). The inputs to the CNN are the module of the Fast Fourier's transform in the range of frequencies between 0 Hz and 25 Hz. We analyzed the discrimination capability of different directions during drawing movements obtaining the best results for X and Y directions. This analysis was performed using a public dataset: Parkinson Disease Spiral Drawings Using Digitized Graphics Tablet dataset. The best results obtained in this work showed an accuracy of 96.5%, a F1-score of 97.7%, and an area under the curve of 99.2%.

5. Parkinson's disease (PD) occurs due to the deficiency of dopamine that regulates various activities of the human body. Researchers have identified that voice is an underlying symptom of PD. Recently, Machine learning (ML) has helped in solving problems of computer vision, natural language processing, speech recognition etc. OBJECTIVES: This paper aims to analyse the effect of feature type selection i.e. MFCC and TQWT on the efficiency of voice based PD detection system along with the use an ensemble learning based classifier for this task. Hence, in the work, various machine learning models, including Logistic Regression, Naive Bayes, KNN,Random Forest, Decision Tree, SVM, MLP, and XG Boost, have been employed and explored for PD detection purpose. The task of Feature selection was also done using minimum-Redundancy and Maximum-Relevance (mRMR) and Recursive Feature Elimination (RFE) techniques

2.1 Existing Problem

Instead of going to hospital and taking MRI scan the existing problem helps the patient to detect the Parkinsons at home with the some basic values extracted from voice recording which is the simple and easiest way.

2.2 References

1. Parkinson Disease Detection Using Various Machine Learning Algorithms Publisher: IEEE By Kanakaprabha S, Arulprakash P, Srikanth R

2. Machine Learning Approaches for Parkinson's Disease Detection Gunjan Pahuja and T. N. Nagabhushan .Published on 22 Oct 2018

3. Efficient detection of Parkinson's disease using deep learning techniques over medical data by Lipsita Sahu, Rohit Sharma, Raghvendra Kumar

4. Parkinson's Disease Detection from Drawing Movements Using Convolutional Neural Networks August 2019 by Manuel Gil-MartIN, Juan M Montero, Ruben San-Segundo

5. Parkinson Disease through Ensemble Machine Learning Approach: A Performance Study Published on July 2018 By EAI Endorsed Transactions on Pervasive Health and Technology - Iqra Nissar, Danish Raza Rizvi, Sarfaraz Masood, Aqib Mir

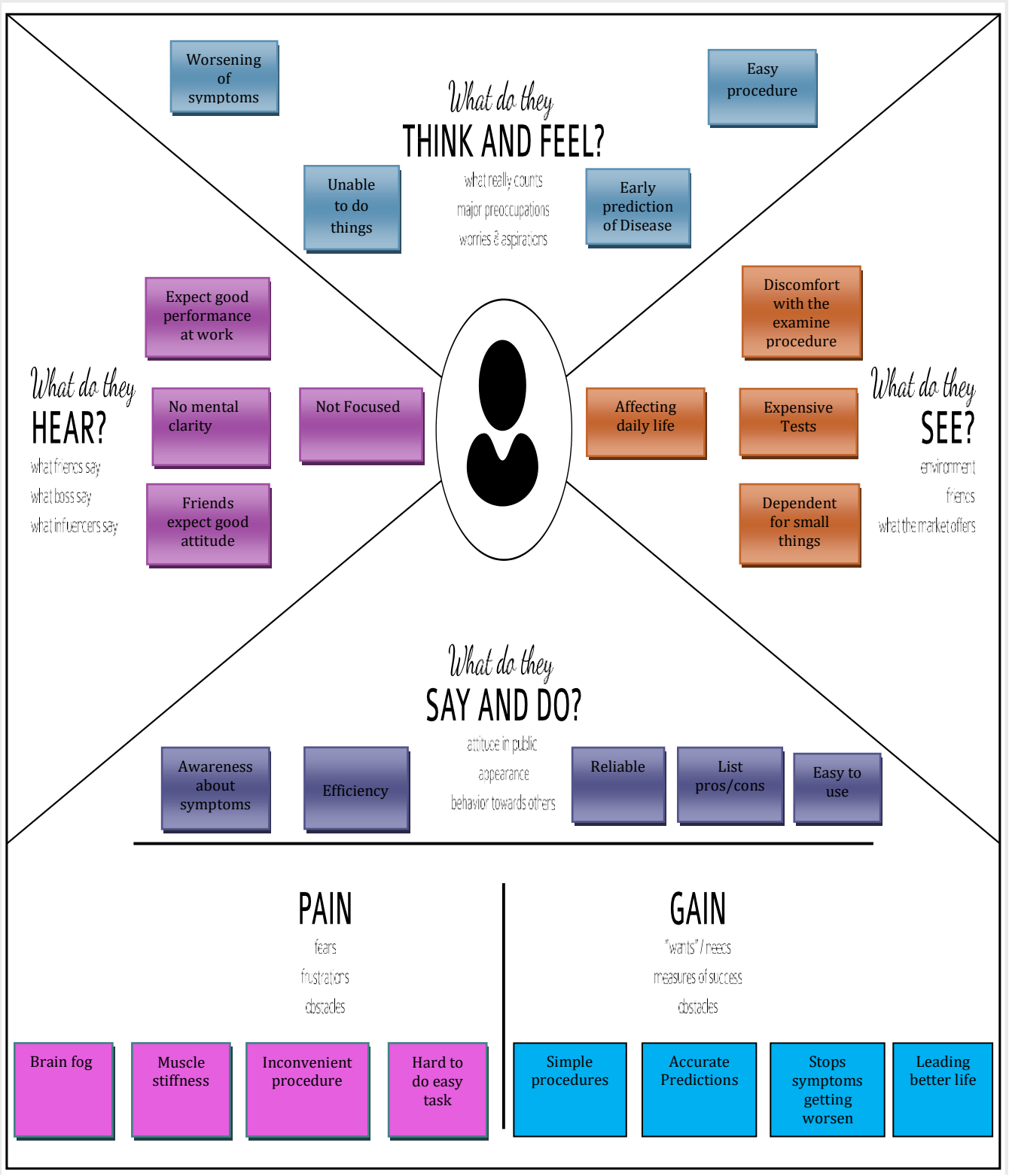
2.3 Problem Statement Defintion

Parkinson's disease (PD) is a common, neurodegenerative disorder, recognized by the motor symptoms of bradykinesia, tremor, rigidity, and postural impairment. At clinical onset, extensive amounts of dopaminergic neurons have already been lost. The duration of this prodromal phase is uncertain, and it is thought to include predominantly non-motor symptoms. The progressive nature and the symptoms of PD are disabling and reduces the quality of life. Among patients affected in working age, early cessation of employment is common, and such socioeconomic consequences of PD may contribute to an impaired quality of life.

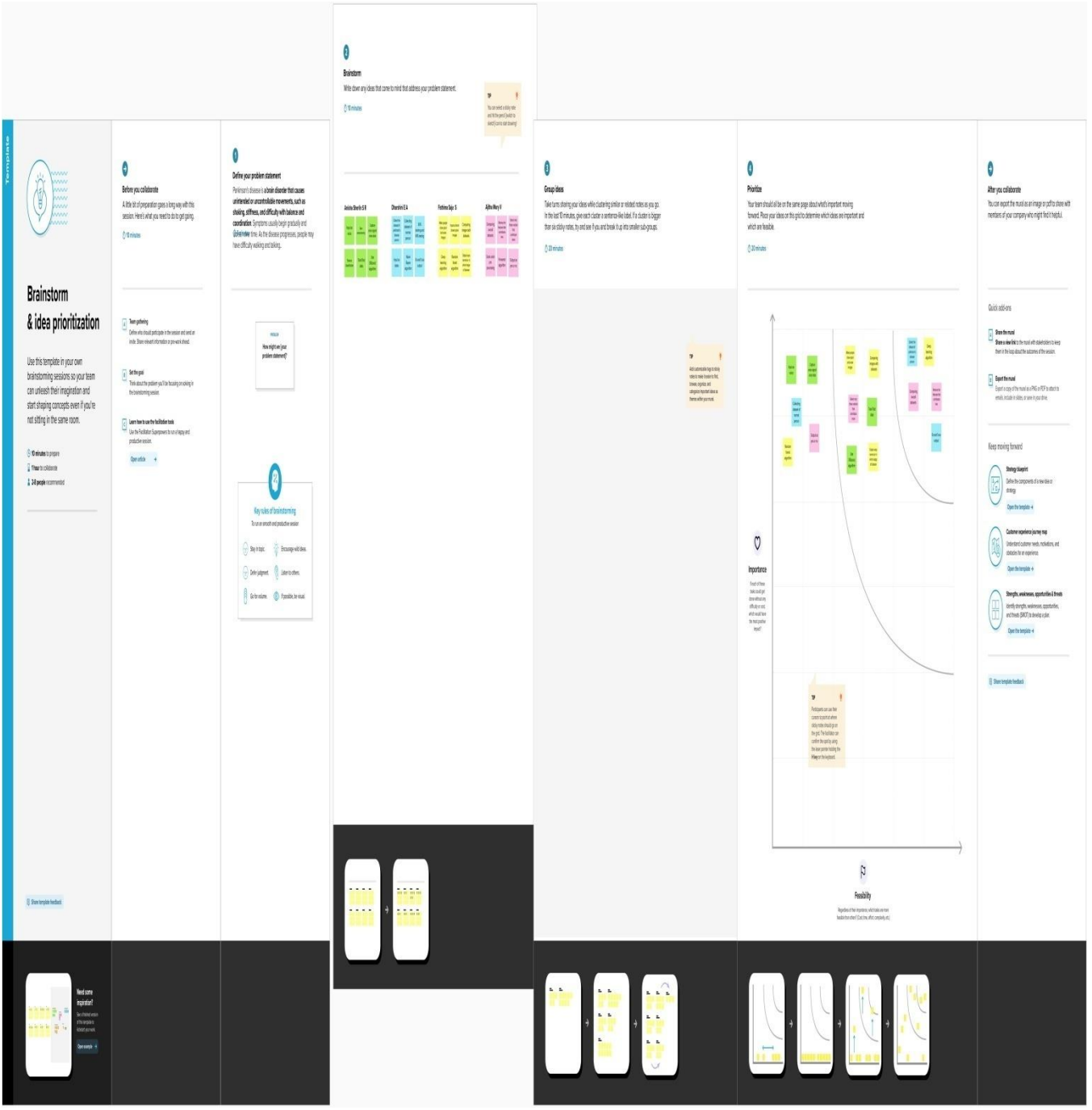
CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



3.3Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	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. As the 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 pronounciation of a short sentence, signal will be recorded through channel simultaneously through mobile phone or microphone. Parkinson disease affect vocal chord 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 behaviours (ICBs). Both patient groups were examined "on" and "off" dopaminergic medication in a counterbalanced order and their behaviour 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 detect without cost and helps to avoid travelling and time.
5	Business Model (Revenue Model)	A free platform with useful feature. Any adult and young people 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 Problems 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

People above the age of 50.

.

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.

Expensive exams.MRI facility are available Only in big hospitals and it's hard for backward people to detect their symptoms the right time

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 notetaking

Parkinsons disease can't be cured, but medications can help control the symptoms, often dramatically. In some more advanced cases, surgery may be advised. Your health care provider may also recommend lifestyle changes, especially ongoing aerobic exercise

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.

Early prediction
Easy procedure
Accurate prediction
Low cost

9. PROBLEM ROOT CAUSE RC

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

Parkinson's disease is caused by a loss of nerve cells in the part of the brain called the substantia nigra. It is an age-related degenerative brain condition, meaning it causes parts of your brain to deteriorate. It's best known for causing slowed movements, Tremors, balance problems and more. Most Cases happen for unknown reasons, but some are inherited.

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)

Patients with Parkinson's disease can experience several behavioral symptoms, such as apathy, agitation, hyper-sexuality stereotypic movements, Pathological gambling, abuse of anti-parkinsonian drugs, and REM sleep behavioral disorders.

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

Identify strong TR & EM

3. TRIGGERS

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

Hard for people to go through an MRI. They wanted to get accurate results by an easy procedure. They triggered by watching advertisement on various websites related to Parkinson's disease.

4. EMOTIONS: BEFORE / AFTER

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.

Customers cannot do their task properly as they have symptoms of Parkinson's disease. After detecting the disease, customers can take treatment and prevent the symptoms get worsen and show a betterment in their performance

10. YOUR SOLUTION

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.

Voice signals are taken from the person and it is
Compared with the dataset of the Parkinson disease person dataset and the disease is detected

8. CHANNELS of BEHAVIOUR

ONLINE

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

OFFLINE

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

Customer take Parkinson's detection test online after notices some symptoms

Customer take further treatment or medications from hospital

Identify strong TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Link (HTML page)
FR-2	User Confirmation	Confirmation via Email
FR-3	Upload voice as input	Add voice Device or through Drive
FR-4	Microphone on	When the microphone is on it recognize the voice and extract some values and detect to return the required output.

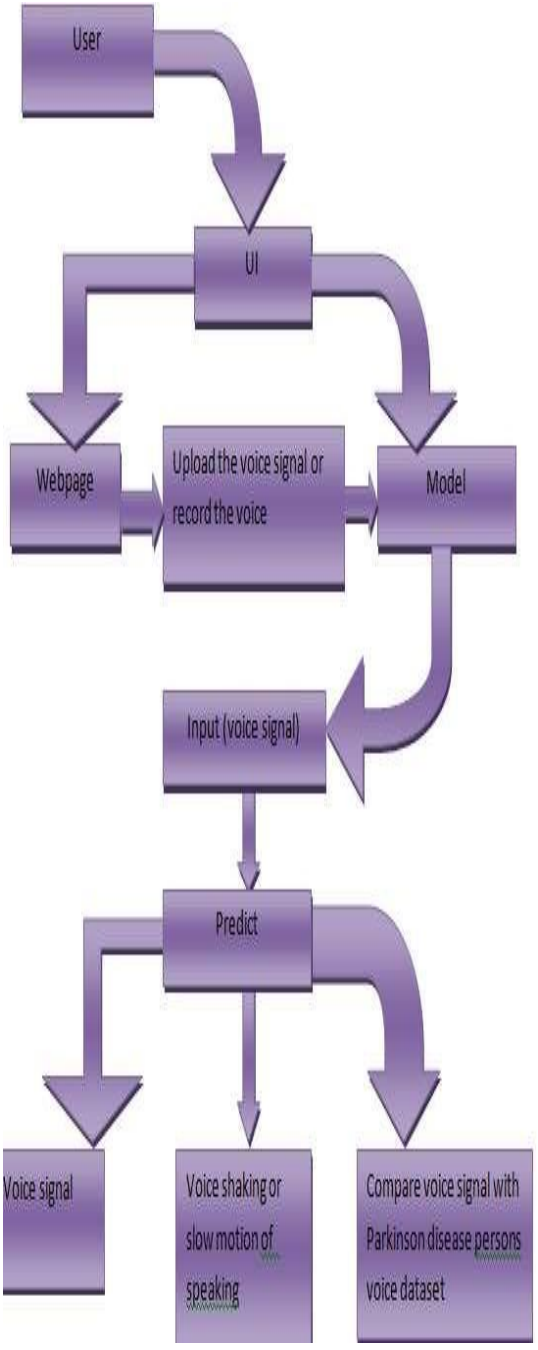
4.2 Non-Functional Requirements

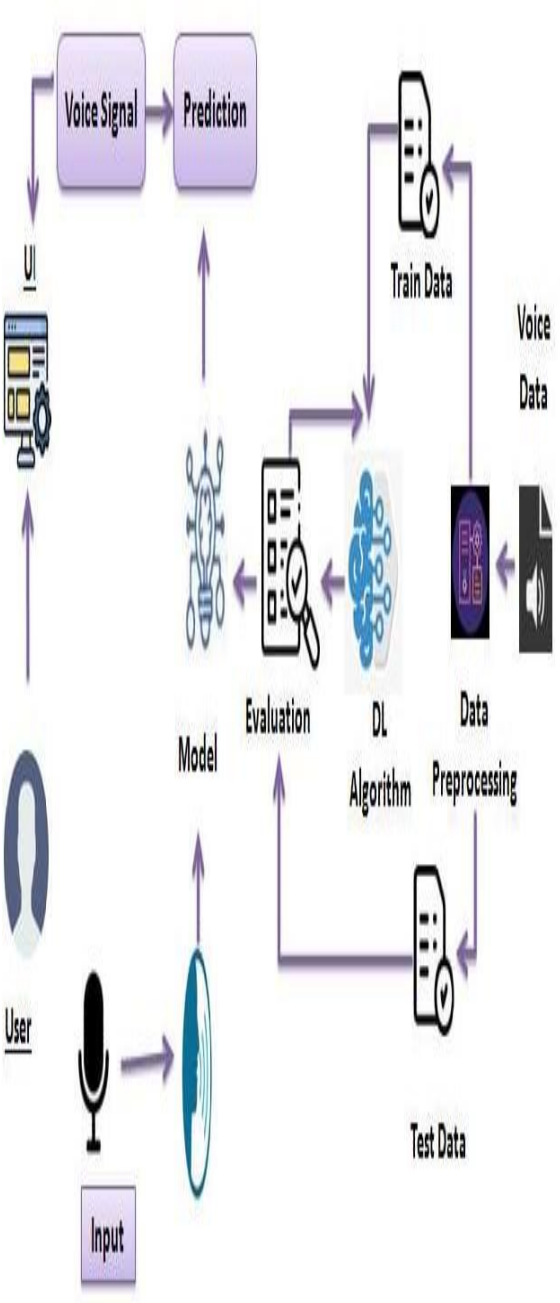
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This software will be easy to use for all users with minimal instructions. User experiences are assessed in a mixed methods approach with patients and experts.
NFR-2	Security	The user of the system should be provided the surety that their account details are secure. The System will provide security against cross site request forgery.
NFR-3	Reliability	This software will be operable in all conditions. Regardless of the person physically challenged(who can’t speak)
NFR-4	Performance	This software will minimize the number of calculations used to perform with more accuracy and processed fast.
NFR-5	Availability	This software will be available to all operating system. While it is currently has a relatively limited role in direct patient care, its evolving role in Complex clinical decision making.
NFR-6	Scalability	This software will be enterprise scalability of AI development and deployment.

CHAPTER 5

PROJECT DESIGN

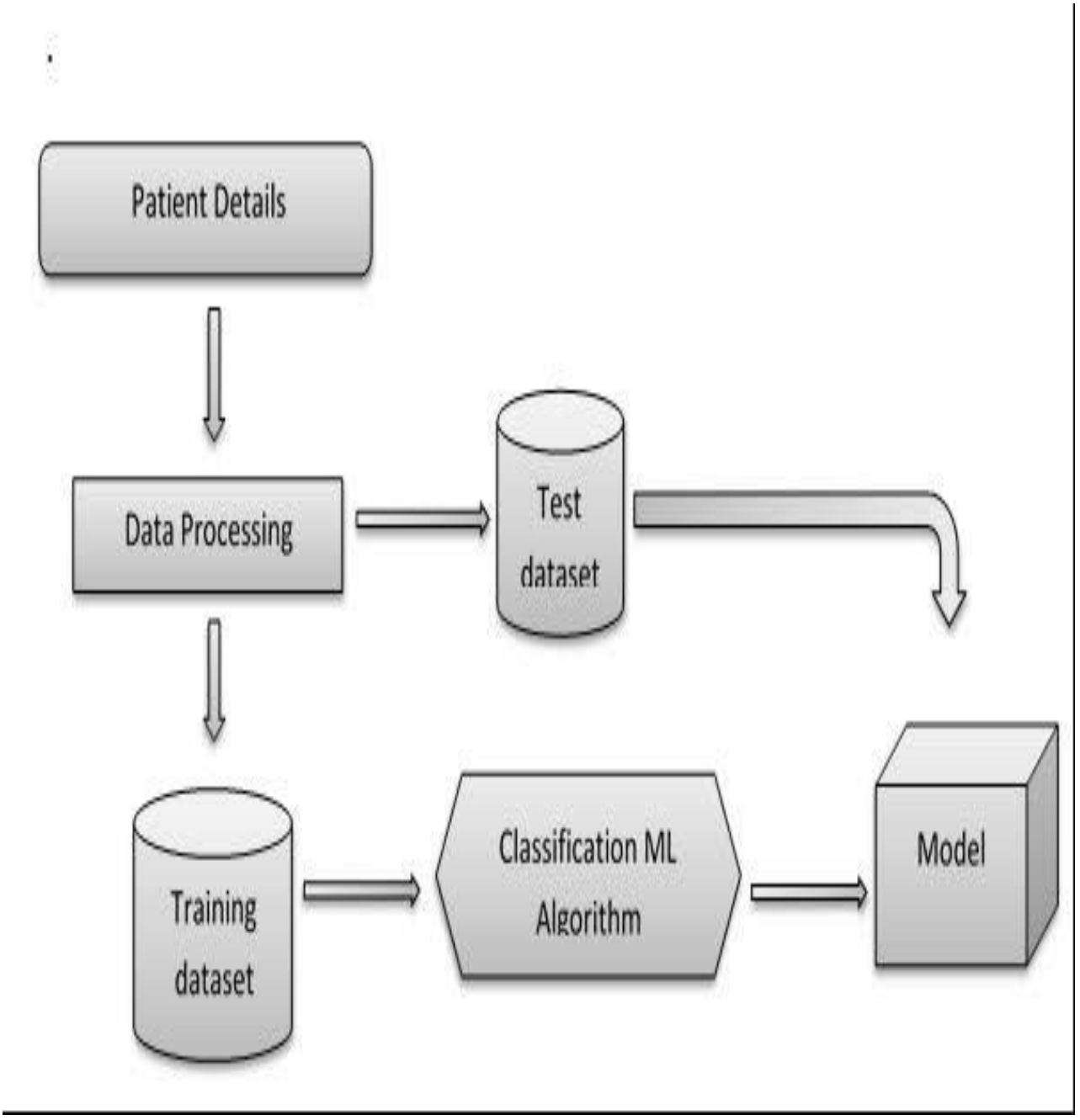
5.1 Data flow diagram



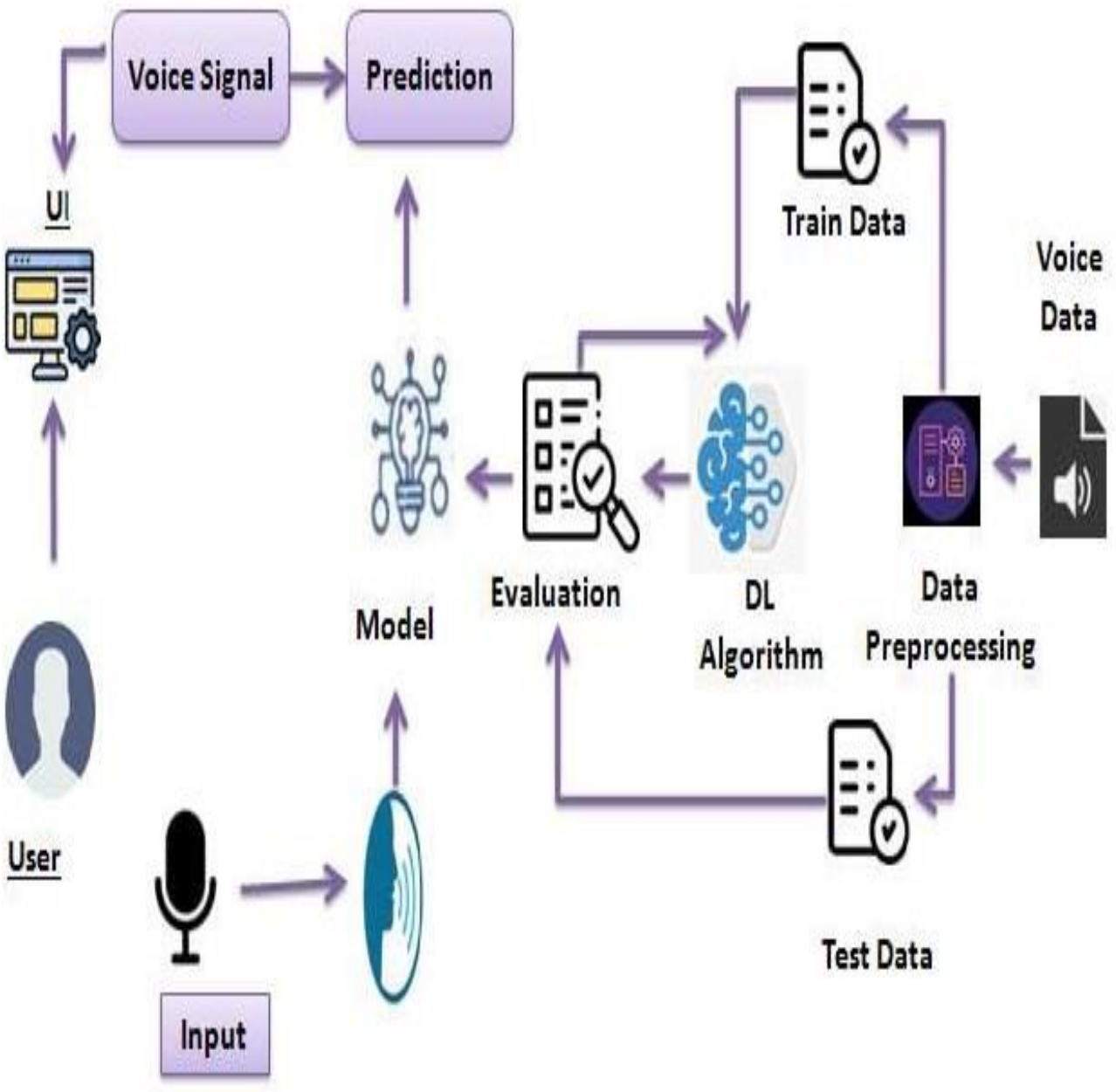


5.2 Solution &Technical Architecture

Solution Architecture



Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Login	USN-1	Entering Web page	Enter the application	High	Sprint 1
	Homepage	USN-2	Entering to the "Homepage" of theUI (Webpage)	Enter the homepage	High	Sprint 1
	About	USN-3	I can click on the "About" to details about the Application	Get the details about the application	Low	Sprint 2
	Begin	USN-4	As a user I can get my voice signal values from the computer.	Choose my voice Recording from my Device and extract the values	High	Sprint 2
	Predict	USN-5	As a user I can turn on the microphone or earphone to record my voice and extract needed values from it	Turn on the microphone or earphone to record the voice and extract values from the recording	High	Sprint 3
		USN-6	Predicting by using voice signal values	Can monitor change in voice or voice shaking and predict parkinsons disease	High	Sprint 3

CHAPTER 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-1	Registration	USN-1	As a user, I can register for the application by entering my username, email, password, contact number and confirming my password.	5	High	TM-1 TM-2
Sprint-1	Login	USN-2	As a user, I can enter the username andpassword after registration for login	5	High	TM-1 TM-2
Sprint-2	Dashboard	USN-3	As a user, I can register for the application through Gmail and see the details in Dashboard	10	Low	TM-3TM-4
Sprint-1	Details about	USN-4	As a user, I can register for the applicationthrough Gmail	5	Medium	TM-1 TM-2
Sprint-1	Login and repeated	USN-5	As a user, I can log into the application by entering email and password	5	High	TM-1 TM-2
Sprint-2	Web page details	USN-6	As a user I must extract certain values fromthe recorded voice and fill the form to detect Parkinsons Disease	10	High	TM-3 TM-4
Sprint-3	Upload the voice signal extracteddetails in the web application	USN-7	As a user I must receive a correct predictedoutput	20	High	TM-1 TM-2
Sprint-4	Provide efficient customer support	USN-8	As a user, I need to get support from developers in case of queries and failure ofservice provided	10	Medium	TM-3 TM-4
Sprint-4	Overview the entire process. Take all the responsibility and act bridge between users and developers	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errorsare fixed	10	High	TM-3 TM-4

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity

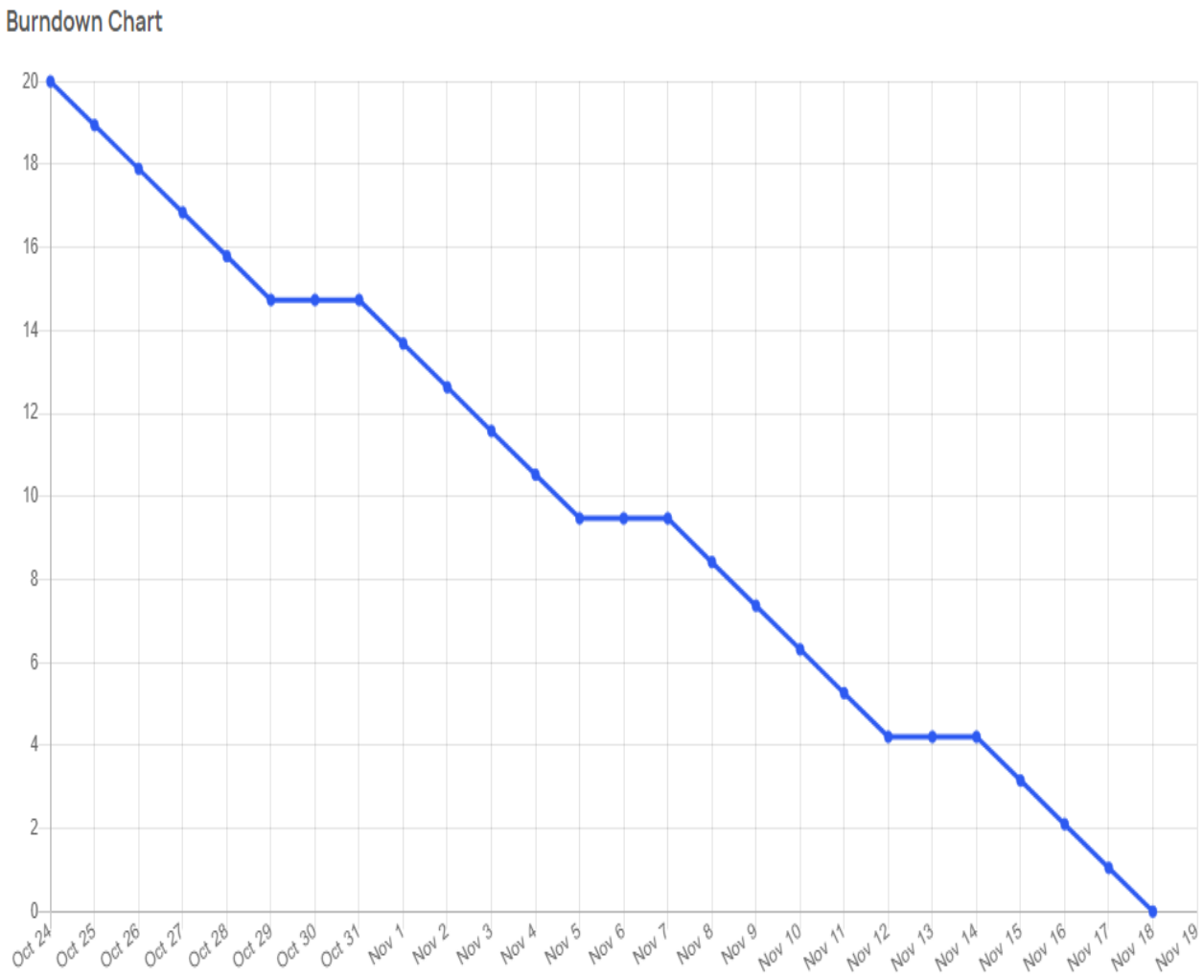
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

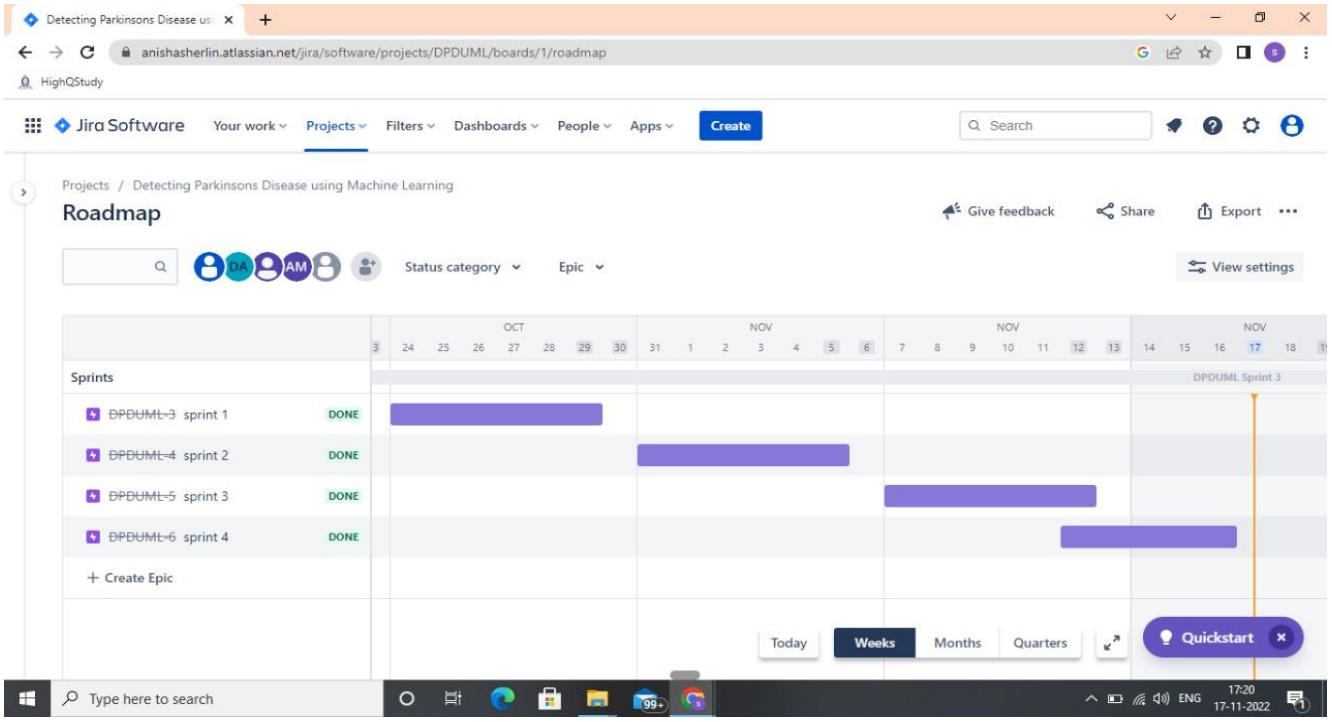
AV = sprint duration / velocity = 20/6 = 3.33

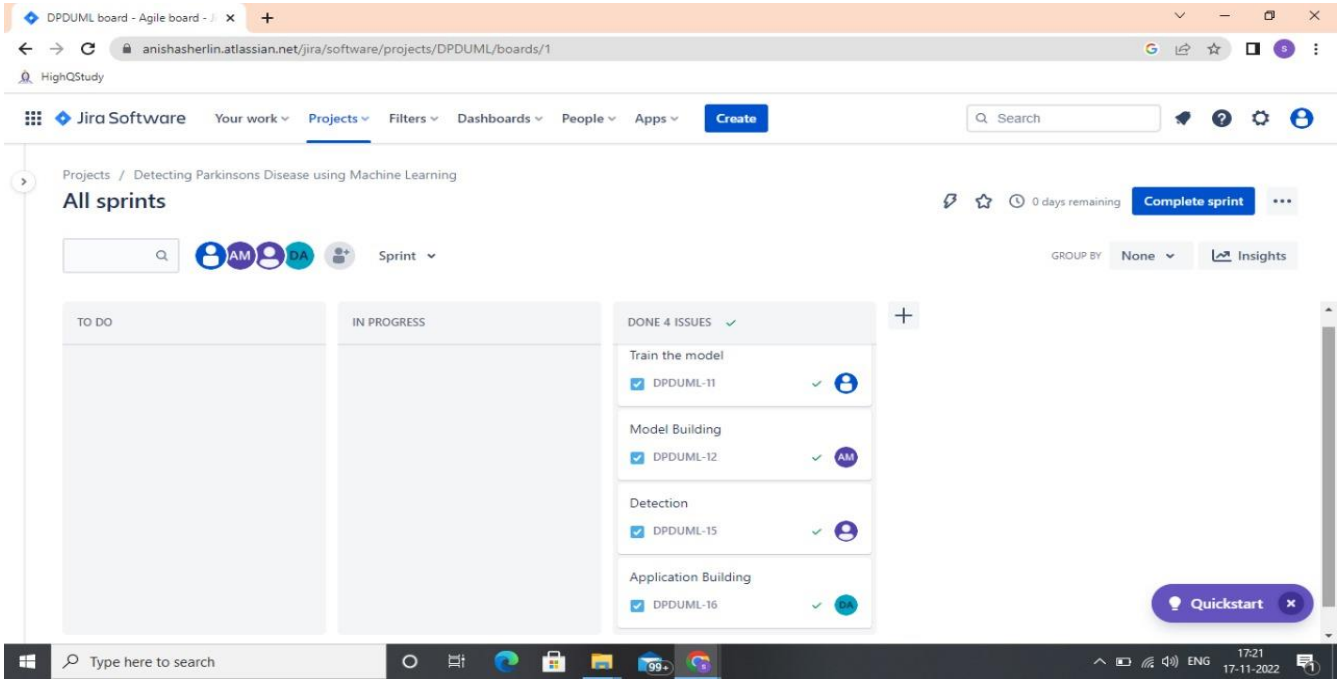
Burn down Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 Reports from JIRA





CHAPTER 7

CODING & SOLUTIONING

7.1 Feature

Home page:

Once the app opens the home page shows the Parkinsons Detection Logo

Login page:

The login page shows after the home page and ask the user for username and password to login

Preview page:

The preview page ask for the voice signal values to upload.

Result page:

The Result page ask for voice signal values and once the click tests is choosen then the app detect the values and tells you whether the person has parkinsons disease or not

7.2 Code

Main.css

```
body {
  background: Linen;
  margin-top: 50px;
  margin-left: 100px;
  }

h1 {
  font-family: Verdana, Geneva, sans-serif;
  font-size: 2.5em;
  color: Black;
  }

p {
  font-family: Georgia, serif;
  font-size: 1.2em;
  color: DarkSlateGray;
  }
.input-form button[type="submit"]
{
        position: relative;
        display: block;
        padding: 7px 39px 18px 39px;
        color: #FFF;
        margin: 0 auto;
```

```
        background: black;
        font-size: 24px;
        text-align: center;
        font-style: normal;
        border: 1px solid #16a085;
        margin-bottom: 10px;
}
.input-form button[type="submit"]:hover
{
        background: #FFA500;
}
```

main1.css

```
body, html {
    margin: 0;
    font-family: sans-serif;
}

.content {
    margin: 0 auto;
    width: 400px;
}

table, td, th {
    border: 1px solid #aaa;
}

table {
    border-collapse: collapse;
    width: 100%;
}

th {
    height: 30px;
}

td {
    text-align: center;
    padding: 5px;
}

.form {
    margin-top: 20px;
}
```

```
#content {
    width: 70%;
}
```

login.css

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);
@importurl('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400
;500;600;700&display=swap');
.body{
  background-image: url('background2.jpeg');
  background-repeat: no-repeat;
  background-size: cover;
}
.login-page {
  width: 360px;
  padding: 8% 0 0;
  margin: auto;
}
.form {
  position: relative;
  z-index: 1;
  background: #48c9b0;
  max-width: 360px;
  margin: 0 auto 100px;
  padding: 45px;
  text-align: center;
  box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}
.form input {
  font-family: FontAwesome, "Roboto", sans-serif;
  outline: 0;
  background: #f2f2f2;
  width: 100%;
  border: 0;
  margin: 0 0 15px;
  padding: 15px;
  box-sizing: border-box;
  font-size: 14px;
border-radius:10px;
}
.form button {
  font-family: "Titillium Web", sans-serif;
  font-size: 14px;
  font-weight: bold;
  letter-spacing: .1em;
```

```
  outline: 0;
  background: #17a589;
  width: 100%;
  border: 0;
border-radius:30px;
  margin: 0px 0px 8px;
  padding: 15px;
  color: #FFFFFF;
  -webkit-transition: all 0.3 ease;
  transition: all 0.3 ease;
  cursor: pointer;
  transition: all 0.2s;
}
.form button:hover,.form button:focus {
  background: #148f77;
    box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
  transform: translateY(-4px);
}
.form button:active {
  transform: translateY(2px);
  box-shadow: 0 2.5px 5px rgba(0, 0, 0, 0.2);
}
.form .message {
  margin: 6px 6px;
  color: #808080;
  font-size: 11px;
  text-align: center;
  font-weight: bold;
  font-style: normal;
}
.form .message a {
  color: #FFFFFF;
  text-decoration: none;
  font-size: 13px;
}
.form .register-form {
  display: none;
}
.container {
  position: relative;
  z-index: 1;
  max-width: 300px;
  margin: 0 auto;
}
```

```
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
  text-align: center;
}
.container .info h1 {
  margin: 0 0 15px;
  padding: 0;
  font-size: 36px;
  font-weight: 300;
  color: #1a1a1a;
}
.container .info span {
  color: #4d4d4d;
  font-size: 12px;
}
.container .info span a {
  color: #000000;
  text-decoration: none;
}
.container .info span .fa {
  color: #EF3B3A;
}
body {
  background: #76b852; /* fallback for old browsers */
  background: -webkit-linear-gradient(right, #76b852, #8DC26F);
  background: -moz-linear-gradient(right, #76b852, #8DC26F);
  background: -o-linear-gradient(right, #76b852, #8DC26F);
  background: linear-gradient(to left, #76b852, #8DC26F);
  font-family: "Roboto", sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
}
::selection{
  color: #000;
```

```
  background: #fff;
}
nav{
  position: fixed;
  width: 100%;
  padding: 10px 0;
  z-index: 12;
}
nav .menu{
  max-width: 1250px;
  margin: auto;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 20px;
}
.menu .logo a{
  text-decoration: none;
  color: #fff;
  font-size: 35px;
  font-weight: 600;
}
.menu ul{
  display: inline-flex;
}
.menu ul li{
  list-style: none;
  margin-left: 7px;
}
.menu ul li:first-child{
  margin-left: 0px;
}
.menu ul li a{
  text-decoration: none;
  color: #fff;
  font-size: 18px;
  font-weight: 500;
  padding: 20px 15px;
  border-radius: 5px;
  transition: all 0.3s ease;
}
.menu ul li a:hover{
  background: #fff;
  color: black;
```

```
}
.center{
  position: absolute;
  top: 52%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 100%;
  padding: 0 20px;
  text-align: center;
}
.center .title{
  color: #fff;
  font-size: 55px;
  font-weight: 600;
}
.center .sub_title{
  color: #fff;
  font-size: 52px;
  font-weight: 600;
}
.center .btns{
  margin-top: 20px;
}
.center .btns button{
  height: 55px;
  width: 170px;
  border-radius: 5px;
  border: none;
  margin: 0 10px;
  border: 2px solid white;
  font-size: 20px;
  font-weight: 500;
  padding: 0 10px;
  cursor: pointer;
  outline: none;
  transition: all 0.3s ease;
}
.center .btns button:first-child{
  color: #fff;
  background: none;
}
.btns button:first-child:hover{
  background: white;
  color: black;
}
```

```
.center .btns button:last-child{
  background: white;
  color: black;
}

h1{
  height:200px;
  margin:0px;
}
```

index.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Parkinson Disease</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css')}}">
    <script src="https://kit.fontawesome.com/a076d05399.js"></script>
   </head>
<body>
  <nav>
    <div class="menu">
      <div class="logo">
        <a href="home">Parkinson Disease</a>
      </div>
      <ul>
        <li><a href="home">Home</a></li>
        <li><a href="login">Login</a></li>
      </ul>
    </div>
  </nav>
  <div class="img"></div>
  <div class="center">
    <div class="title">Parkinson Disease Predictor</div>
    <div class="sub_title">Using Machine Learning </div>
<!--  <div class="btns">-->
<!-- <button>Learn More</button>-->
<!--  <button>Subscribe</button>-->
<!--  </div>-->
  </div>
</body>
</html>
```

show_csv_data.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href='/static/css/main.css' />
  </head>
  <body>
    <h1>Flask GET, POST tutorial</h1>
    <p>Display csv file in html table Flask</p>
{{ data_var|safe }}
    </body>
</html>
```

upload.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href='/static/css/main.css' />
    <link rel="stylesheet" href="{{ url_for('static', filename='css/login_style.css')}}">
  </head>
  <body
  <nav>
    <div class="menu">
      <div class="logo">
          </div>
      <ul>
        <li><a href="home">Home</a></li>
          <li><a href="login">Login</a></li>
      </ul>
    </div>
  </nav>
    <h1>UPLOAD</h1>
    <form method="POST" enctype="multipart/form-data" action="/">
        <input type="file" id="myFile" name="uploaded-file" accept=".csv">
        <input type="submit" value="Upload file">
    </form>
```

```
  <br>
  <p style="color:blue;">Choose csv file to upload</p>
  <form action="/show_data" >
       <input type="submit" value="Preview" />
    </form>
  </body>
</html>
```

upload2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href='/static/css/main.css' />
    <link rel="stylesheet" href="{{ url_for('static', filename='css/login_style.css')}}">
  </head>
  <body>
  <nav>
    <div class="menu">
      <div class="logo">
           </div>
      <ul>
        <li><a href="home">Home</a></li>
          <li><a href="login">Login</a></li>
      </ul>
    </div>
  </nav>
    <h1>UPLAOD</h1>
    <form method="POST" enctype="multipart/form-data" action="/">
       <input type="file" id="myFile" name="uploaded-file" accept=".csv">
       <input type="submit" value="Upload file">
    </form>
  <br>
  <p style="color:green;">file uploaded successfully</p>
```

```
  <form action="/show_data">
       <input type="submit" value="Preview" />
    </form>
  {{ data_var|safe }}
  </body>
</html>
```

CHAPTER 8

TESTING

8.1 Test Cases

Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N	BUG ID	Executed By
user is able to see the login	1.Enter URL and click go 2. User to login	http://127.0.0.1:5000	Login should display	Working as	Pass				
Verify the UI elements in Login	1.Enter URL and click go 2.Verify login with below UI elements: b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	http://127.0.0.1:5000	Application should show below UI elements: a.username text box b.password text box c.Login button with green colour d.New customer? Create account link e.Last password? Recovery	Working as expected	Fail	Steps are not clear to follow		BUG-1234	
Verify user is able to log into application with Valid credentials	1.Enter URL(http://127.0.0.1:5000) and click go. 2.Enter Valid username/email in Email text box 3.Enter valid password in password text box 4.Click on login button	Username: user1 password: 1234	User should navigate to user account homepage	Working as expected	Pass				
Verify user is able to log into application with InValid credentials	1.Enter URL(http://127.0.0.1:5000) and click go 2.Enter InValid username text box 3.Enter valid password in password text box 4.Click on login button	Username: jjjj password: 7588	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
Verify user is able to log into application with InValid credentials	1.Enter URL(http://127.0.0.1:5000) and click go 2.Enter Valid username/email in Email text box 3.Enter Invalid password in password text box 4.Click on login button	Username: aaaa password: 6876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
Verify user is able to log into application with InValid credentials	1.Enter URL(http://127.0.0.1:5000) and click go 2.Enter InValid username in text box 3.Enter Invalid password in password text box 4.Click on login button	Username: bbbb password:19788	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				

8.2 User Acceptance Testing

- Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of the [Product Name] project at the time of the release to User Acceptance Testing (UAT).

- Defect Analysis:

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	12	3	4	3	22
Duplicate	2	0	3	0	5
External	3	2	0	2	7
Fixed	10	1	3	21	35
Not Reproduced	0	0	1	0	1
Skipped	0	1	1	1	3
Won't Fix	0	4	1	2	7
Totals	27	11	13	29	80

- Test Case Analysis

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

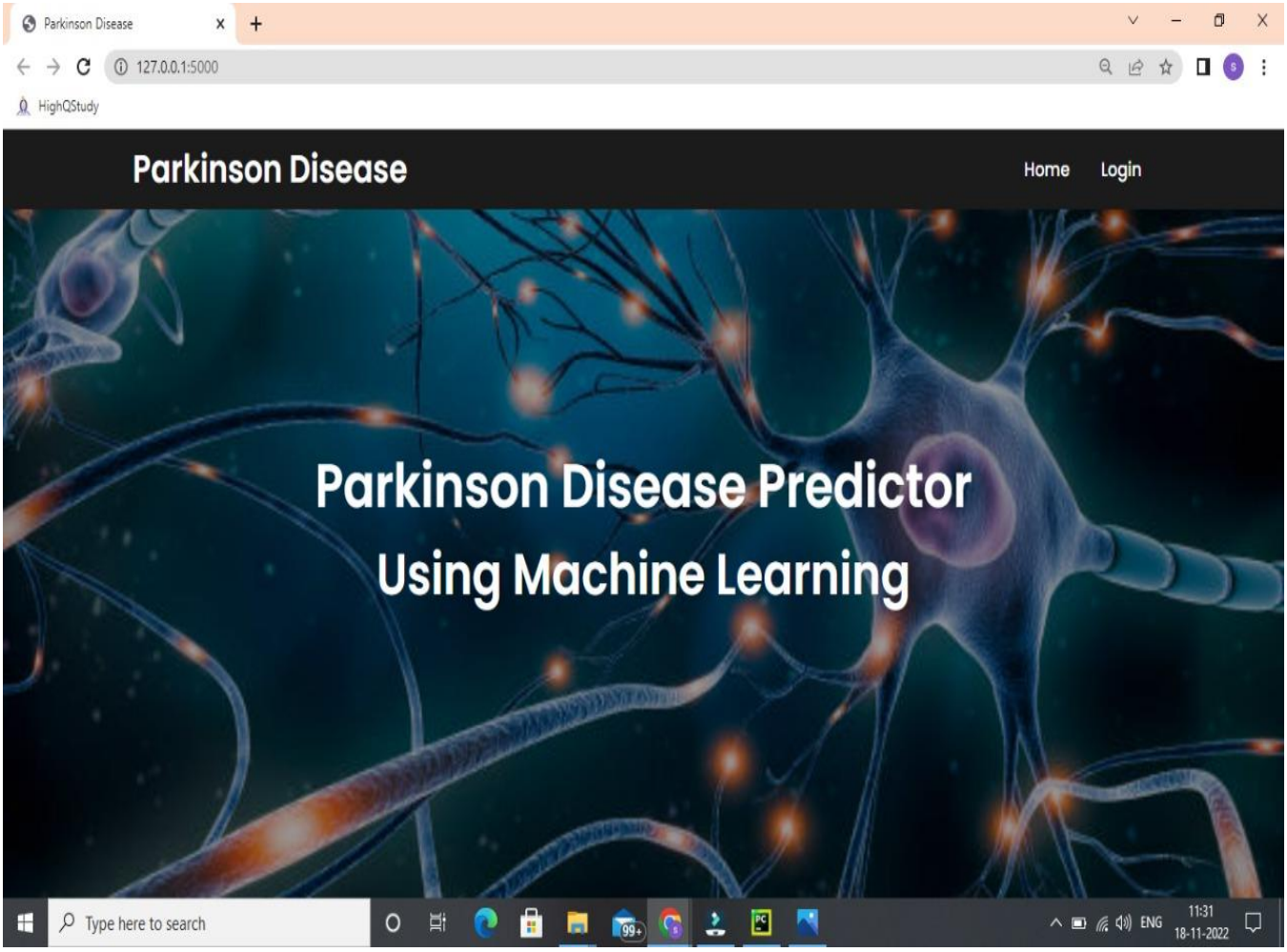
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	53	0	0	53
Security	2	0	0	2
Outsource Shipping	4	0	0	4
Exception Reporting	8	0	0	8
Final Report Output	5	0	0	5
Version Control	2	0	0	2

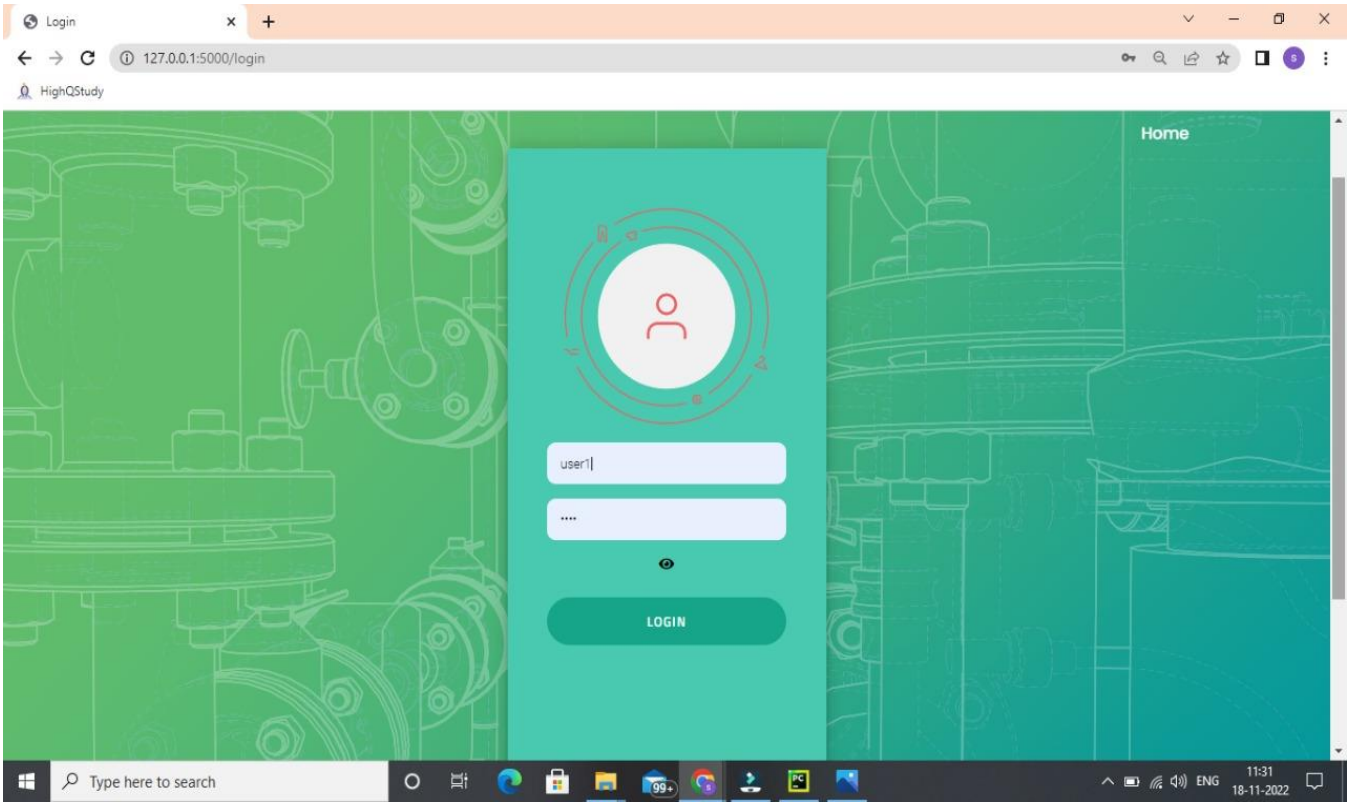
CHAPTER 9

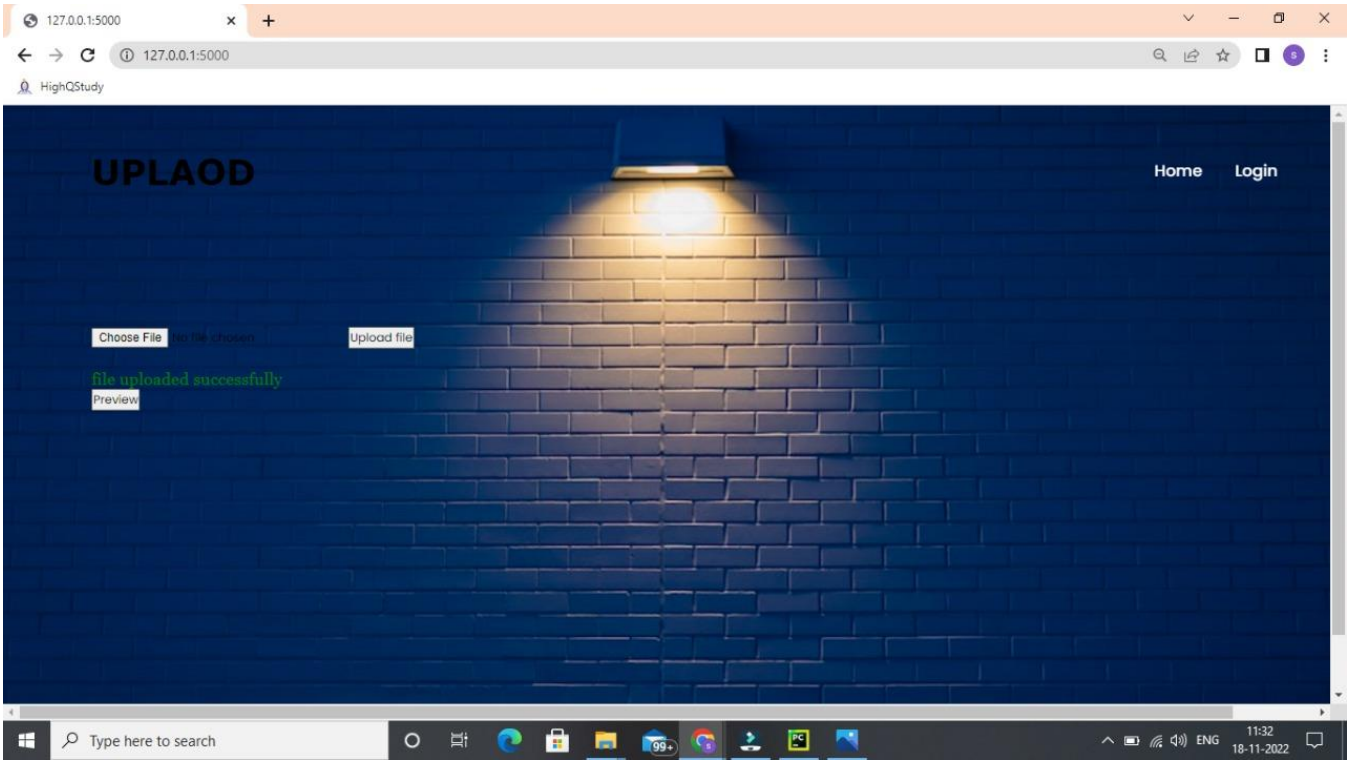
RESULTS

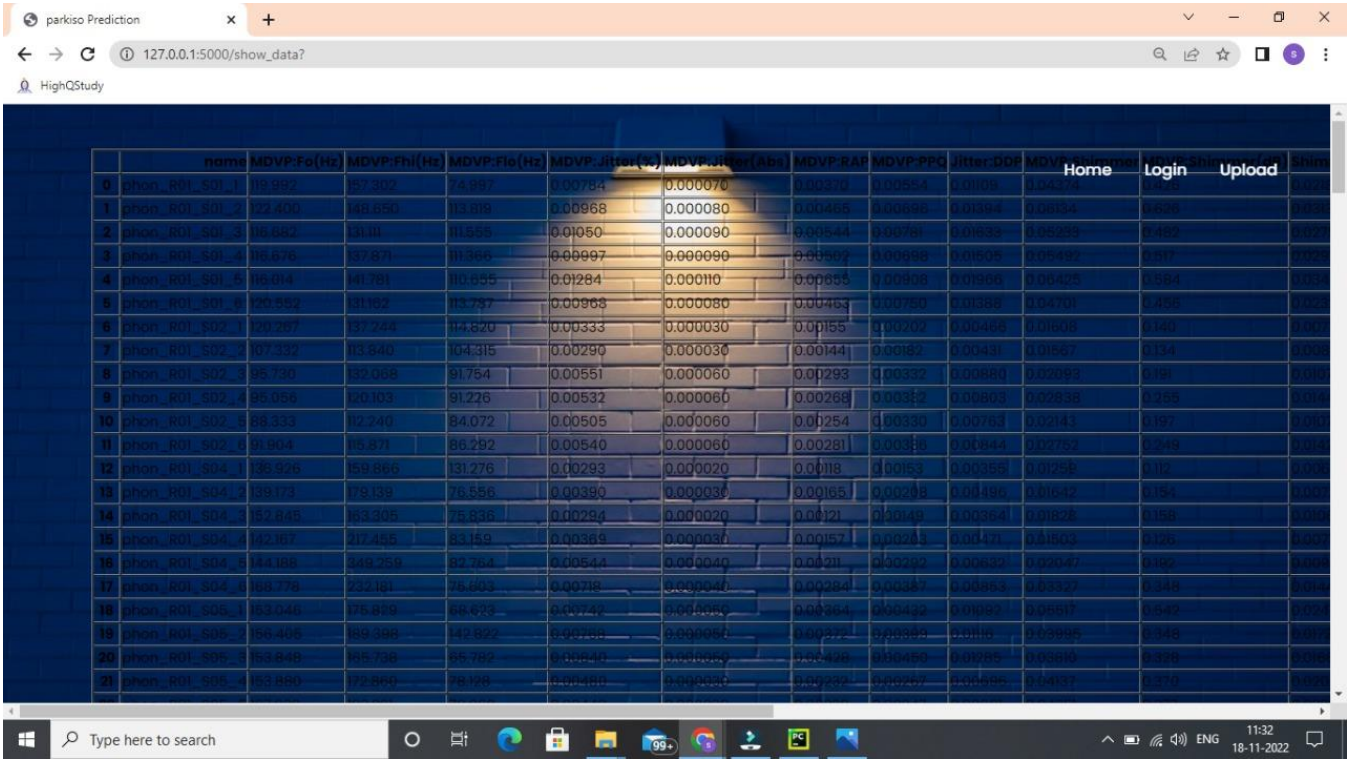
In this project, we found that Parkinsons disease can be detected using the value's obtained from voice recording.

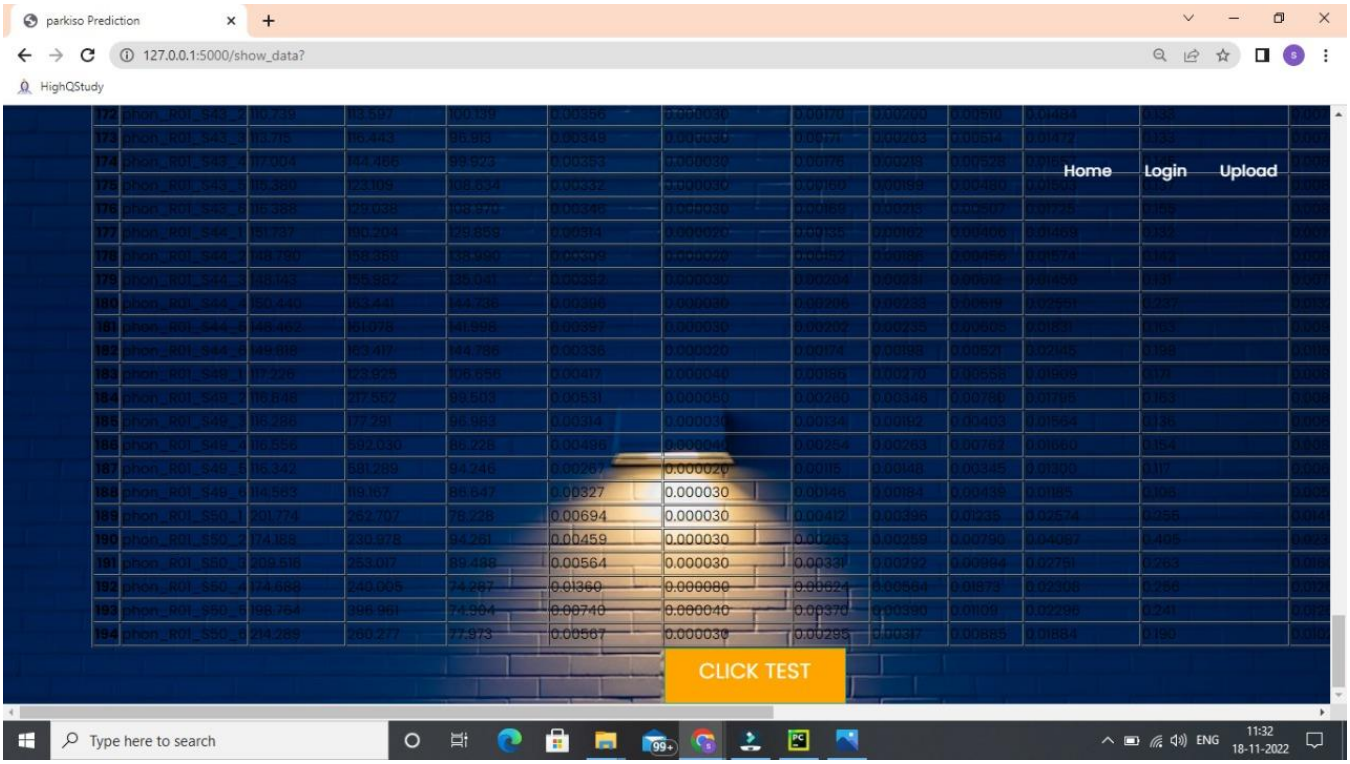
Final findings (Output) of the project along the screenshots as follows.

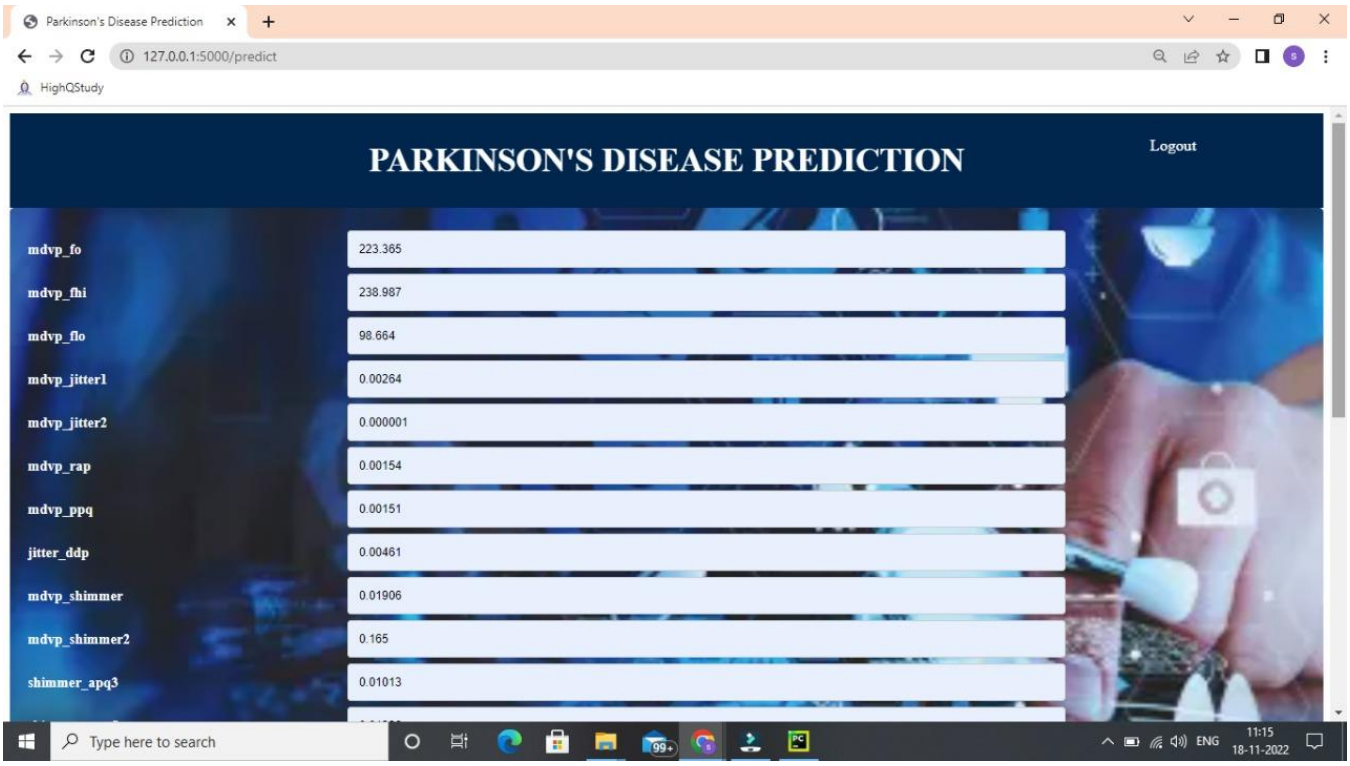


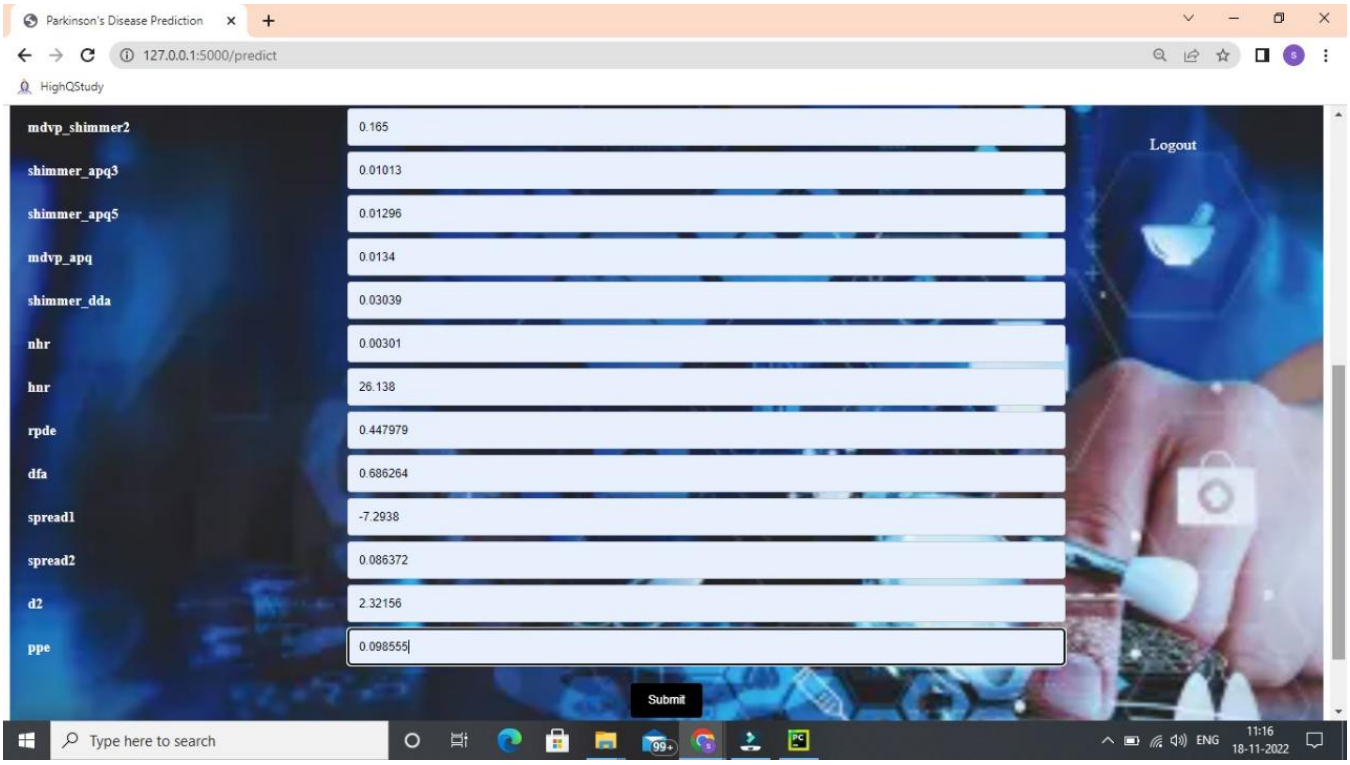

Login
127.0.0.1:5000/login
HighQStudy
Home
user1
LOGIN
Type here to search
ENG
11:31
18-11-2022

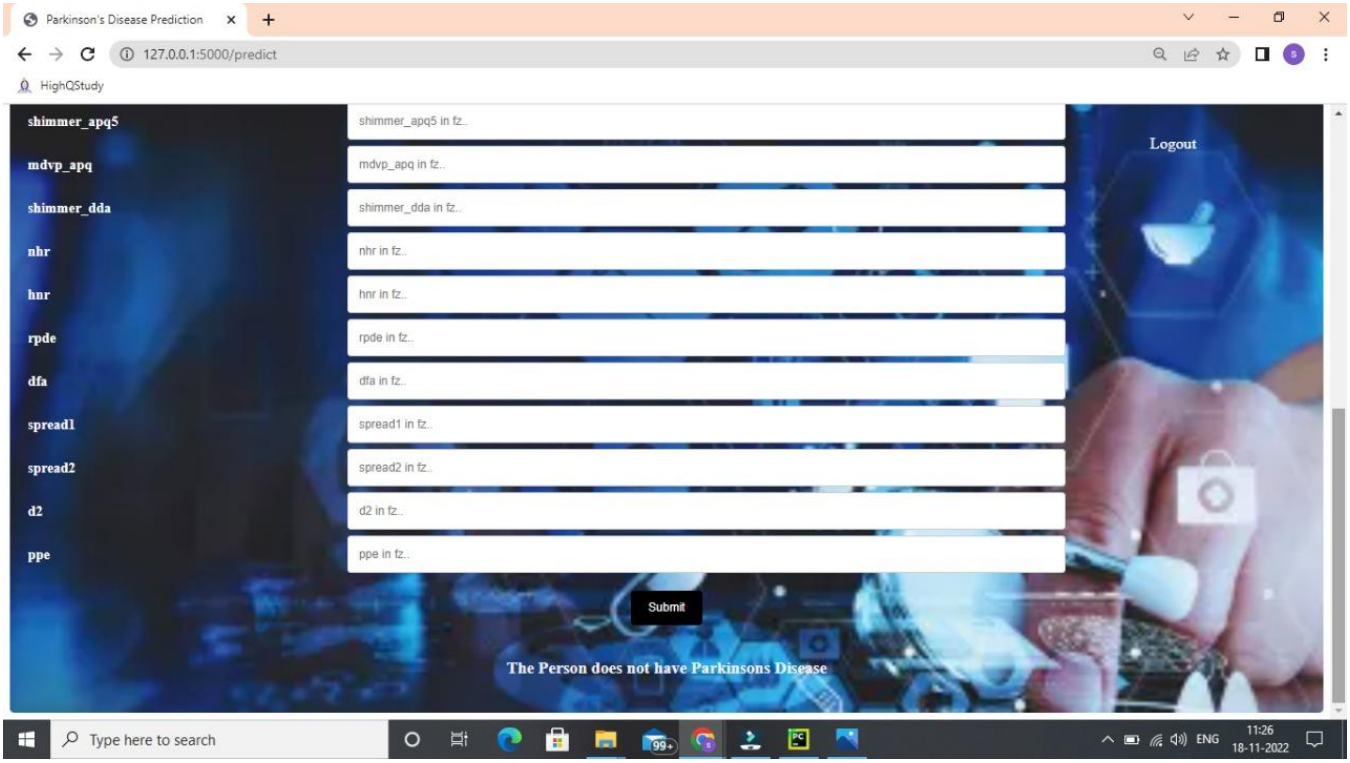

127.0.0.1:5000
127.0.0.1:5000
HighQStudy
UPLAOD
Home
Login
Choose File
Upload file
file uploaded successfully
Preview
Type here to search
ENG
11:32
18-11-2022

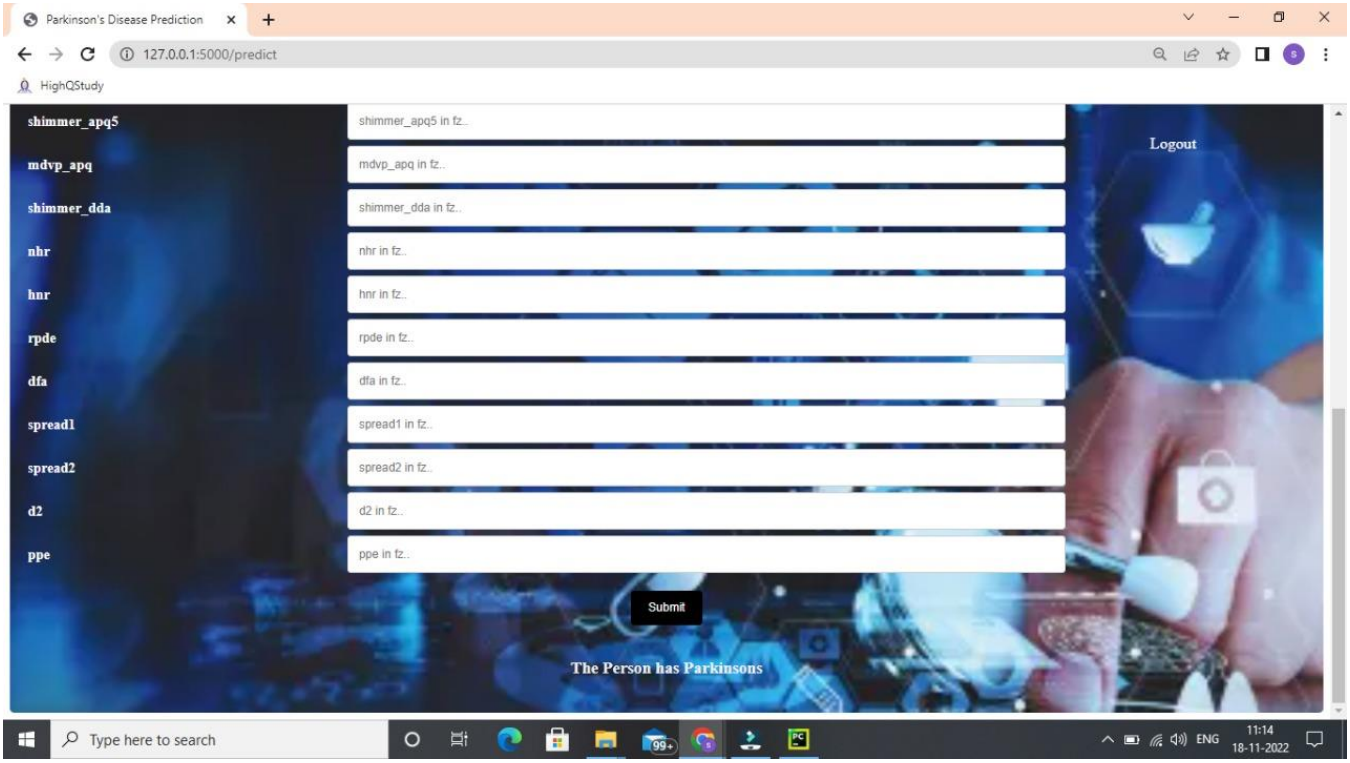

parkiso Prediction
127.0.0.1:5000/show_data?
HighQStudy
Home
Login
Upload
Type here to search
ENG
11:32
18-11-2022


parkiso Prediction
127.0.0.1:5000/show_data?
HighQStudy
Home
Login
Upload
CLICK TEST
Type here to search
ENG
11:32
18-11-2022

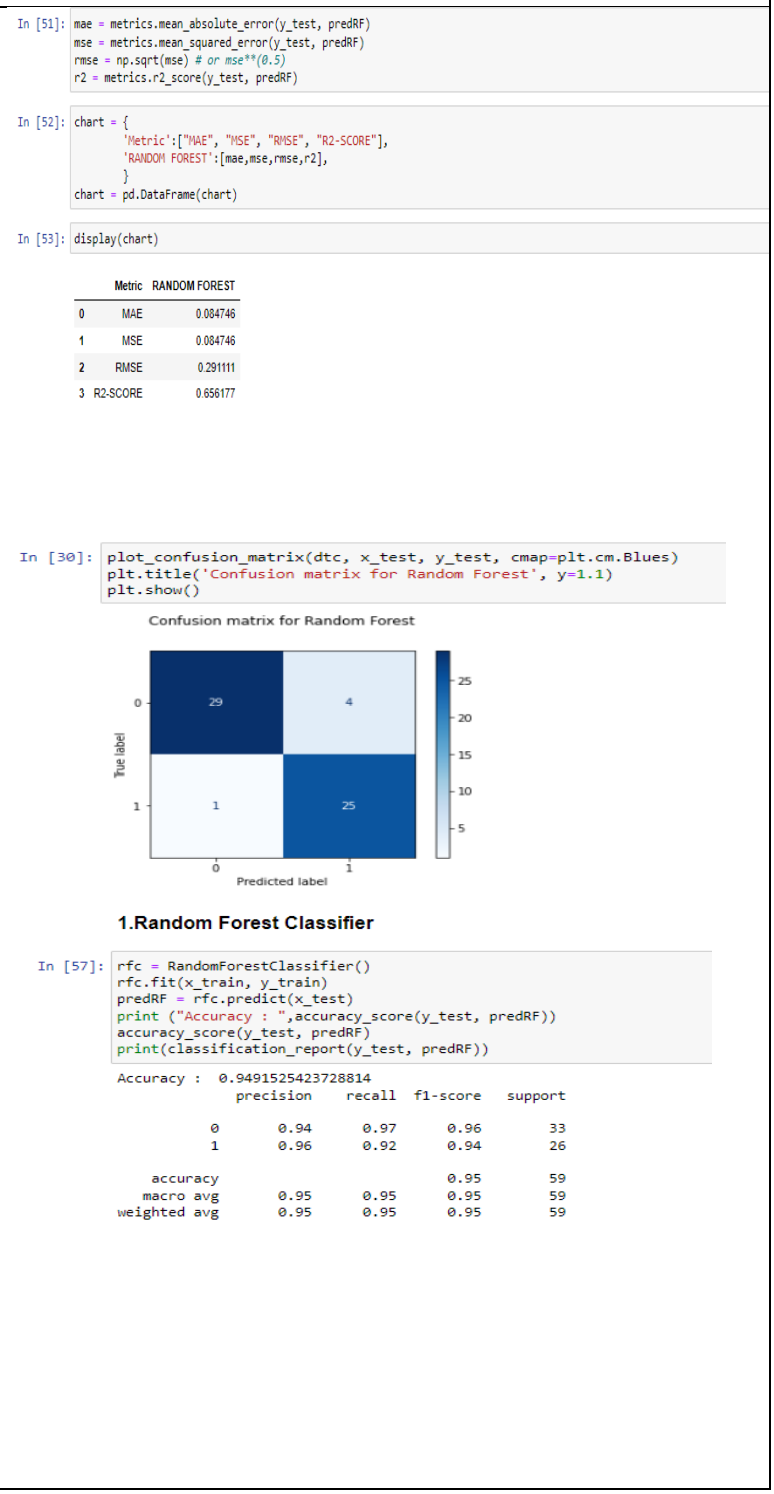

Parkinson's Disease Prediction
127.0.0.1:5000/predict
HighQStudy
PARKINSON'S DISEASE PREDICTION
Logout
mdvp_fo 223.365
mdvp_fhi 238.987
mdvp_flo 98.664
mdvp_jitter1 0.00264
mdvp_jitter2 0.000001
mdvp_rap 0.00154
mdvp_ppq 0.00151
jitter_ddp 0.00461
mdvp_shimmer 0.01906
mdvp_shimmer2 0.165
shimmer_apq3 0.01013
Type here to search
ENG
11:15
18-11-2022


Parkinson's Disease Prediction
127.0.0.1:5000/predict
HighQStudy
mdvp_shimmer2 0.165
shimmer_apq3 0.01013
shimmer_apq5 0.01296
mdvp_apq 0.0134
shimmer_dda 0.03039
nhr 0.00301
hnr 26.138
rpde 0.447979
dfa 0.686264
spread1 -7.2938
spread2 0.086372
d2 2.32156
ppe 0.098555
Logout
Submit
Type here to search
ENG
11:16
18-11-2022


Parkinson's Disease Prediction
127.0.0.1:5000/predict
HighQStudy
shimmer_apq5
shimmer_apq5 in fz..
mdvp_apq
mdvp_apq in fz..
shimmer_dda
shimmer_dda in fz..
nhr
nhr in fz..
hnr
hnr in fz..
rpde
rpde in fz..
dfa
dfa in fz..
spread1
spread1 in fz..
spread2
spread2 in fz..
d2
d2 in fz..
ppe
ppe in fz..
Submit
Logout
The Person does not have Parkinsons Disease
Type here to search
ENG
11:26
18-11-2022


Parkinson's Disease Prediction
127.0.0.1:5000/predict
HighQStudy
shimmer_apq5
shimmer_apq5 in fz..
mdvp_apq
mdvp_apq in fz..
shimmer_dda
shimmer_dda in fz..
nhr
nhr in fz..
hnr
hnr in fz..
rpde
rpde in fz..
dfa
dfa in fz..
spread1
spread1 in fz..
spread2
spread2 in fz..
d2
d2 in fz..
ppe
ppe in fz..
Submit
Logout
The Person has Parkinsons
Type here to search
ENG
11:14
18-11-2022

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE -0.084746 , MSE - 0.084746 , RMSE - 0.291111 , R2 score - 0.656177 **Classification Model:** Confusion Matrix, Accuray Score- 0.9491525423728814 & Classification Report	

```
In [51]: mae = metrics.mean_absolute_error(y_test, predRF)
         mse = metrics.mean_squared_error(y_test, predRF)
         rmse = np.sqrt(mse) # or mse**(0.5)
         r2 = metrics.r2_score(y_test, predRF)

In [52]: chart = {
                'Metric':["MAE", "MSE", "RMSE", "R2-SCORE"],
                'RANDOM FOREST':[mae,mse,rmse,r2],
                }
         chart = pd.DataFrame(chart)

In [53]: display(chart)
```

	Metric	RANDOM FOREST
0	MAE	0.084746
1	MSE	0.084746
2	RMSE	0.291111
3	R2-SCORE	0.656177

```
In [30]: plot_confusion_matrix(dtc, x_test, y_test, cmap=plt.cm.Blues)
         plt.title('Confusion matrix for Random Forest', y=1.1)
         plt.show()
```


1.Random Forest Classifier

```
In [57]: rfc = RandomForestClassifier()
         rfc.fit(x_train, y_train)
         predRF = rfc.predict(x_test)
         print ("Accuracy : ",accuracy_score(y_test, predRF))
         accuracy_score(y_test, predRF)
         print(classification_report(y_test, predRF))
```

```
Accuracy :  0.9491525423728814
              precision    recall  f1-score   support

           0       0.94      0.97      0.96        33
           1       0.96      0.92      0.94        26

    accuracy                           0.95        59
   macro avg       0.95      0.95      0.95        59
weighted avg       0.95      0.95      0.95        59
```

2.	Tune the Model	Hyperparameter Tuning Validation Method – [0.95744681 0.91489362 0.93617021 0.91489362 0.85106383]	In [58]: ``` from sklearn.model_selection import cross_val_score, StratifiedKFold skf = StratifiedKFold(n_splits=5, shuffle= True, random_state= 17) ``` In [60]: ``` val_scores = cross_val_score(estimator= rfc, X= x_train, y= y_train, cv= skf) ``` In [61]: ``` avg_score=val_scores.mean() ``` In [62]: ``` print ("Cross Validation Scores : ",val_scores) print ("Average CV Score : ",avg_score) print ("Number of CV Scores used in Average : ",len(val_scores)) ``` ``` Cross Validation Scores : [0.95744681 0.91489362 0.93617021 0.91489362 0.85106383] Average CV Score : 0.9148936170212766 Number of CV Scores used in Average : 5 ```

CHAPTER 10

ADVANTAGES &DISADVANTAGES

Advantages:

- Major advantage of this tool is that it helps to detect the Parkinsons disease from home.
- It is also easy to use and is quicker to detect Parkinsons disease.
- It can also be performed in any place and everywhere.

Disadvantages:

- The person's who doesn't able to speak cannot detect Parkinsons using this tool

CHAPTER 11

CONCLUSION

Parkinson's disease has been plaguing humans for thousands of years and was described in detail in ancient medical writings. Early sufferers from it effects were treated with varying results by a variety of plant-based treatments, some of which are still in use today. With the discovery of dopamine in the twentieth century and the subsequent development of dopamine replacement therapy, plus surgical techniques such as deep brain stimulation (DBS), many of the debilitating symptoms are now successfully treated—at least for a time.

The hope is to find the cause of PD, along with treatments that stop the disease from progressing. Of particular interest, PD research is uncovering what may turn out to be a common pathophysiologic mechanism underlying dementia and PD. For now, healthcare providers must continue to educate themselves about currently available treatments and hope for better alternatives in the near future.

CHAPTER 12

FUTURE SCOPE

- The tool can be made more accurate by adding even more algorithms.
- The tool can be not only detected by voice also by image and also Gait detection.
- Can add and get more personal information and past medical information.
- Can add more security and authentication.

CHAPTER 13

APPENDIX

Source code

main.py

```
from flask import Flask,request, url_for, redirect, render_template,session
from werkzeug.utils import secure_filename
import pickle
import numpy as np
import pandas as pd
import os
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestClassifier
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
UPLOAD_FOLDER = os.path.join('static', 'uploads')
ALLOWED_EXTENSIONS = {'csv'}
parkinsons_data = pd.read_csv('static/uploads/parkinsons.csv')
parkinsons_data.drop(['name'], axis=1, inplace=True)
x = parkinsons_data.drop(['status'], axis=1)
y = parkinsons_data['status']
sm = SMOTE(random_state=300)
x, y = sm.fit_resample(x, y)
scaler = MinMaxScaler((-1, 1))
x = scaler.fit_transform(x)
Y = y
x_train, x_test, y_train, y_test = train_test_split(x, Y, test_size=0.20, random_state=20)
rfc = RandomForestClassifier()
```

```
rfc.fit(x_train, y_train)
filename = 'parkinson_model.pickle'
pickle.dump(rfc, open(filename, 'wb'))
app = Flask(_name_, template_folder='templates', static_folder='static')
# Configure upload file path flask
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.secret_key = 'This is your secret key to utilize session in Flask'
@app.route('/')
def hello_world():
    return render_template("index.html")
@app.route('/home')
def home():
    return render_template("index.html")
@app.route('/login')
def login():
    return render_template("login.html")
@app.route('/form_login',methods=['POST','GET'])
def login1():
    database={'user1':'1234','user2':'abcd','admin':'admin'}
    name1=request.form['username']
    pwd=request.form['password']
    if name1 not in database:
        return render_template('login.html',info='Invalid User')
    else:
        if database[name1]!=pwd:
            return render_template('login.html',info='Invalid password')
        else:
            # return render_template('login.html',info='login Successfull')
            return render_template('upload.html',name=name1)
@app.route('/upload')
```

```
def upload_file():
    return render_template('upload.html')
@app.route('/', methods=("POST", "GET"))
def uploadFile():
    if request.method == 'POST':
        # upload file flask
        uploaded_df = request.files['uploaded-file']
        # Extracting uploaded data file name
        data_filename = secure_filename(uploaded_df.filename)
        # flask upload file to database (defined uploaded folder in static path)
        uploaded_df.save(os.path.join(app.config['UPLOAD_FOLDER'], data_filename))
        # Storing uploaded file path in flask session
        session['uploaded_data_file_path'] = os.path.join(app.config['UPLOAD_FOLDER'],
data_filename)
        return render_template('upload2.html')
@app.route('/show_data')
def showData():
    # Retrieving uploaded file path from session
    data_file_path = session.get('uploaded_data_file_path', None)
    # read csv file in python flask (reading uploaded csv file from uploaded server location)
    uploaded_df = pd.read_csv(data_file_path)
    # pandas dataframe to html table flask
    uploaded_df_html = uploaded_df.to_html()
    return render_template('preview.html', data_var=uploaded_df_html)
@app.route('/input_data',methods=['GET'])  # route to display the home page
def inputPage():
    return render_template("form1.html")
@app.route('/predict',methods=['POST','GET']) # route to show the predictions in a web UI
def predict():
    if request.method == 'POST':
```

```
    mdvp_fo = float(request.form['mdvp_fo'])
    mdvp_fhi = float(request.form['mdvp_fhi'])
    mdvp_flo = float(request.form['mdvp_flo'])
    mdvp_jitter1 = float(request.form['mdvp_jitter1'])
    mdvp_jitter2 = float(request.form['mdvp_jitter2'])
    mdvp_rap = float(request.form['mdvp_rap'])
    mdvp_ppq = float(request.form['mdvp_ppq'])
    jitter_ddp = float(request.form['jitter_ddp'])
    mdvp_shimmer = float(request.form['mdvp_shimmer'])
    mdvp_shimmer2 = float(request.form['mdvp_shimmer2'])
    shimmer_apq3 = float(request.form['shimmer_apq3'])
    shimmer_apq5 = float(request.form['shimmer_apq5'])
    mdvp_apq = float(request.form['mdvp_apq'])
    shimmer_dda = float(request.form['shimmer_dda'])
    nhr = float(request.form['nhr'])
    hnr = float(request.form['hnr'])
    rpde = float(request.form['rpde'])
    dfa = float(request.form['dfa'])
    spread1 = float(request.form['spread1'])
    spread2 = float(request.form['spread2'])
    d2 = float(request.form['d2'])
    ppe = float(request.form['ppe'])
    input_data = (mdvp_fo, mdvp_fhi, mdvp_flo, mdvp_jitter1, mdvp_jitter2, mdvp_rap, mdvp_ppq,
jitter_ddp, mdvp_shimmer,
        mdvp_shimmer2, shimmer_apq3, shimmer_apq5, mdvp_apq, shimmer_dda, nhr, hnr, rpde, d2,
dfa, spread1,
        spread2, ppe)
    # changing input data to numpy array
    input_data_numpy = np.asarray(input_data)
    # reshaping the numpy array
```

```
        input_data_reshape = input_data_numpy.reshape(1, -1)
       # standardizing the input data
        std_data = scaler.transform(input_data_reshape)
        filename = 'parkinson_model.pickle'
        # loading the model file from the storage
        loaded_model = pickle.load(open(filename, 'rb'))
        # predictions using the loaded model file
        prediction = loaded_model.predict(std_data)
        print('prediction is', prediction)
        if (prediction[0] == 0):
            return render_template('form1.html', prediction_text="The Person does not have Parkinsons Disease")
        else:
            return render_template('form1.html', prediction_text="The Person has Parkinsons")
         else:
            return render_template('form1.html')
        if _name_ == '_main_':
        app.run(debug=True)
```

GitHub Link: IBM-EPBL/IBM-Project-30061-1660139320

Project Demo Link: https://youtu.be/eI8Kh3rpqTo