PROJECT REPORT

DETECTION OF PARKINSON'S DISEASE USING MACHINE LEARNING

SUBMITTED BY

PNT2022TMID23220

Archana Shreee S	**913119205006**
Hailly J	**913119205012**
Maheswari B	**913119205023**
Sindhuja C	**913119205043**

TABLE OF CONTENTS

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 1
    7.2 Feature 2
    7.3 Database Schema (if Applicable)
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

1.1 PROJECT OVERVIEW

Parkinson's disease is a progressive disorder of the central nervous system affecting movement and inducing tremors and stiffness. It has 5 stages to it and affects more than 1 million individuals every year in India. This is chronic and has no cure yet. It is a neurodegenerative disorder affecting dopamine-producing neurons in the brain. For detecting PD, various machine learning models such as logistic regression, naive Bayes, KNN, and forest decision tree were used, with the features used here being minimum-redundancy, maximum-relevance and recursive feature elimination. The accuracy obtained was 95.3% using data from the UCI machine learning repository. The researchers found that the drawing speed was slower and the pen pressure is lower among Parkinson's patients. One of the indications of Parkinson's is tremors and rigidity in the muscles, making it difficult to draw smooth spirals and waves. It is possible to detect Parkinson's disease using the drawings alone instead of measuring the speed and pressure of the pen on paper. Our goal is to quantify the visual appearance(using HOG method) of these drawings and then train a machine learning model to classify them. In this project, We are using Histogram of Oriented Gradients (HOG) image descriptor along with a XGBoost to automatically detect Parkinson's disease in hand-drawn images of spirals and waves.

1.2 PURPOSE

By using machine learning techniques, the problem can be solved with minimal error rate. The voice dataset of Parkinson's disease from the UCI Machine learning library is used as input. Also our proposed system provides accurate results by integrating spiral drawing inputs of normal and Parkinson's affected patients. Machine learning also allows for combining different modalities, such as magnetic resonance imaging (MRI) and single-photon emission computed tomography (SPECT) data. in the diagnosis of PD. By using machine learning approaches, we may therefore identify relevant features that are not traditionally used in the clinical diagnosis of PD and rely on these alternative measures to detect PD in preclinical stages or atypical forms. In recent years, the number of publications on the application of machine learning to the diagnosis of PD has increased. Although previous studies have reviewed the use of machine learning in the diagnosis and assessment of PD, they were limited to the analysis of motor symptoms, kinematics, and wearable sensor data.The application of machine learning to clinical and non-clinical data of different modalities has often led to high diagnostic accuracies in human participants, therefore may encourage the adaptation of machine learning algorithms and novel biomarkers in clinical settings to assist more accurate and informed decision making.While Parkinson's cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

More than 10 million people are living with Parkinson's Disease worldwide, according to the Parkinson's Foundation. While Parkinson's cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life. The researchers found that the drawing speed was slower and the pen pressure is lower among Parkinson's patients. One of the indications of Parkinson's is tremors and rigidity in the muscles, making it difficult to draw smooth spirals and waves. It is possible to detect Parkinson's disease using the drawings alone instead of measuring the speed and pressure of the pen on paper.

2.2 REFERENCES

Diagnosis of Parkinson's Disease using Fuzzy C-Means Clustering and Pattern Recognition

Indira R

The author used fuzzy C-means clustering and pattern recognition based approach for the discrimination between healthy and parkinson disease affected people. The authors of this paper have achieved 68.04% accuracy, 75.34% sensitivity and 45.83% specificity.

Feature Relevance Analysis and Classification of Parkinson Disease TeleMonitoring Data Through Data Mining Techniques

R. Geeta

Tele-monitoring dataset and dataset comparison classes Motor-UPDRS and Total-UPDRS (Unified Parkinson Disease Rating scale). Random tree classification achieved 100% accuracy .

A vision-based analysis system for gait recognition in patients with Parkinson's disease

Cho, C. at al

Cho, C. at al. (2009) proposed system utilizes an algorithm combining principal component analysis (PCA) with linear discriminant analysis (LDA). We propose a gait analysis system which can detect the gait pattern of Parkinson's disease using computer vision. This system comprises three main parts: pre-processing, training and recognition. Experimental results showed that LDA had a recognition rate for Parkinsonian gait of 95.49%.

Design and Analysis of Data Mining Based Prediction Model for Parkinson's disease

Azad, C., et al

Azad, C., et al. (2013) proposed prediction model tree based classification model decision tree, ID3 and decision stumps are used for training and testing the effectiveness many symptoms that lead to Parkinson's disease such age environmental factor, trembling in the legs, arms, hands, impaired speech articulation and production difficulties. Decision tree, ID3 and decision stumps our prediction model provides accuracy 85.08%, 75.33% and 83.55% or classification error 14.92%, 24.67% and 16.45% respectively.

Novel speech signal processing algorithms for high accuracy classification of Parkinson's disease

A.Tsanas et al

A.Tsanas et al. (2011) proposed a nonlinear signal approach large dataset (dataset are voice/speech recorded without requiring physician presence in the clinical) apply a wide range known speech signal algorithm. This paper was performed using a nonlinear regression and classification algorithm, and supported visibility of frequent, remote, cost-effective, accurate UPDRS telemonitoring based on self-administered speech tests.

Automatic recognition of Parkinson's disease from sustained phonation tests using ANN and adaptive neuro-fuzzy classifier

Caglar et al

Caglar et al. (2010) proposed ANN (Two types of the ANNs were used for classification: Multilayer Perceptron (MLP) and Radial Basis Function (RBF) Networks) and Adaptive Neuro-Fuzzy Classifier (ANFC) with linguistic hedges to discriminate between healthy people and people with PD. Adaptive Neuro-Fuzzy Classifiers with linguistic hedges gave the best recognition results with %95.38 training and %94.72 testing classifying performance indeed.

Speech Analysis for Diagnosis of Parkinson's Disease Using Genetic Algorithm and Support Vector Machine

Shahbakhi et al

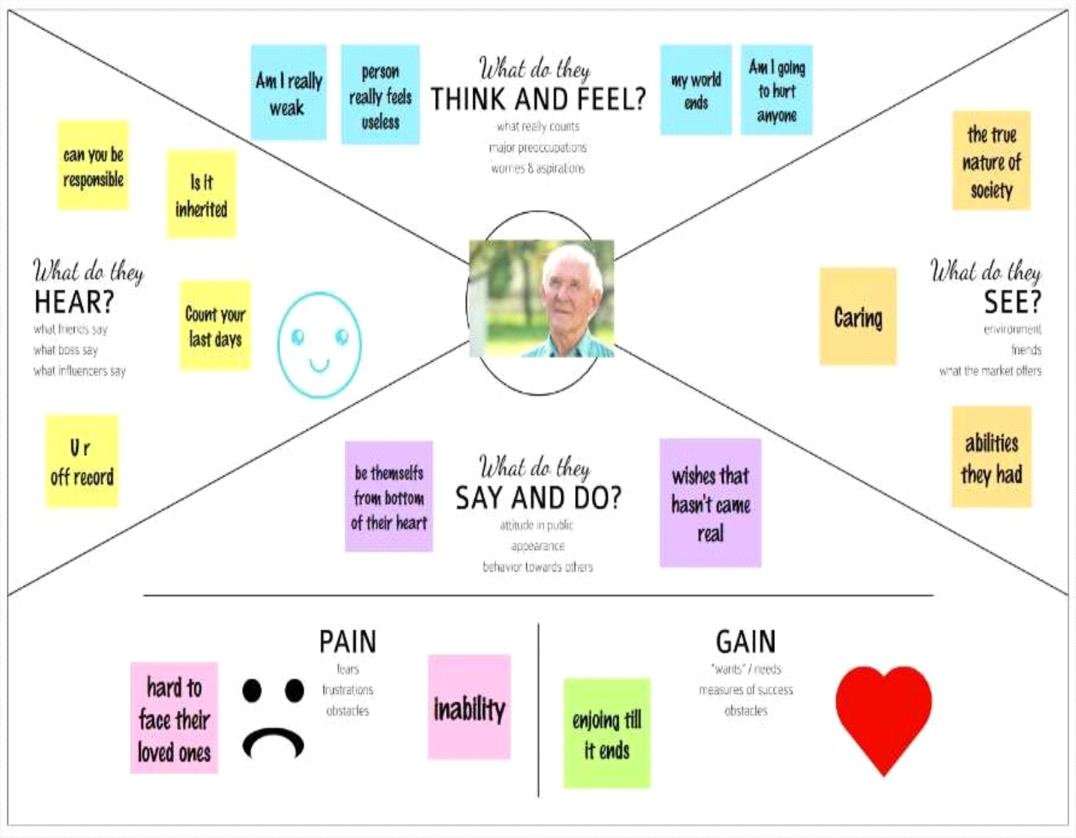
Shahbakhi et al. (2014) presented that a Genetic Algorithm (GA) and SVM were used for classification between healthy and people with Parkinson. Voice signals that 14 features were based on F0 (fundamental frequency or pitch), jitter, shimmer and noise to harmonics ratio, which are main factors in voice signal. Results show that classification accuracy 94.50, 93.66 and 94.22 per 4, 7 and 9 optimized features respectively.

2.3 PROBLEM STATEMENT DEFINITION

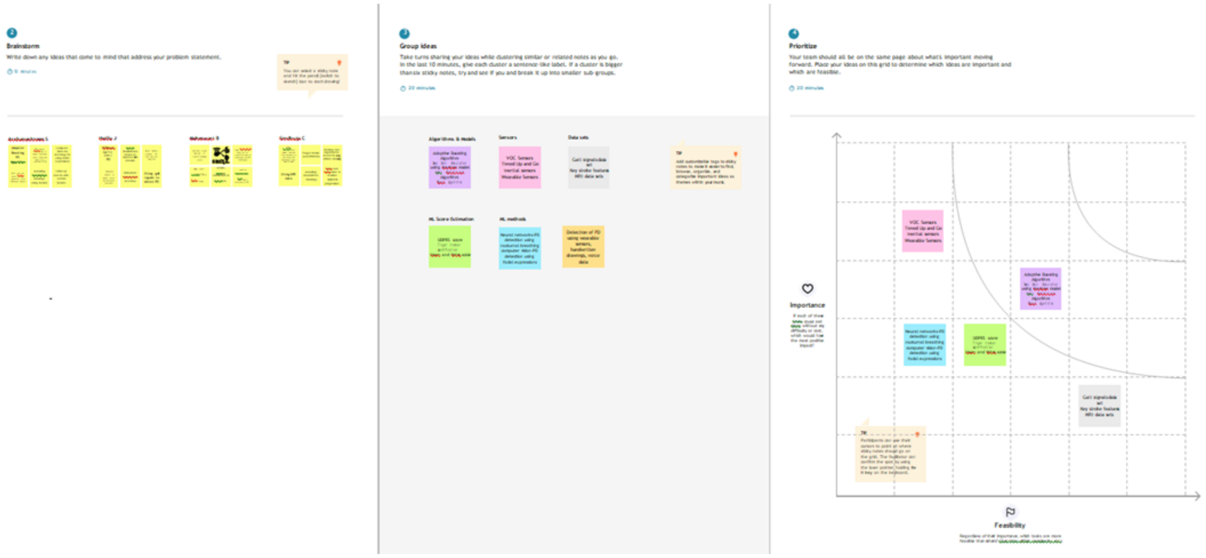
Medical observations and assessment of clinical indicators, including the identification of a variety of motor symptoms, are often used to diagnose Parkinson's disease (PD). Traditional diagnostic procedures, on the other hand, may be vulnerable to subjectivity because they rely on the assessment of motions that are sometimes subtle to human sight and hence difficult to define, potentially leading to misdiagnosis. Meanwhile, early nonmotor symptoms of Parkinson's disease can be minor and be caused by a variety of other illnesses. As a result, these symptoms are frequently missed, making early PD diagnosis difficult.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING

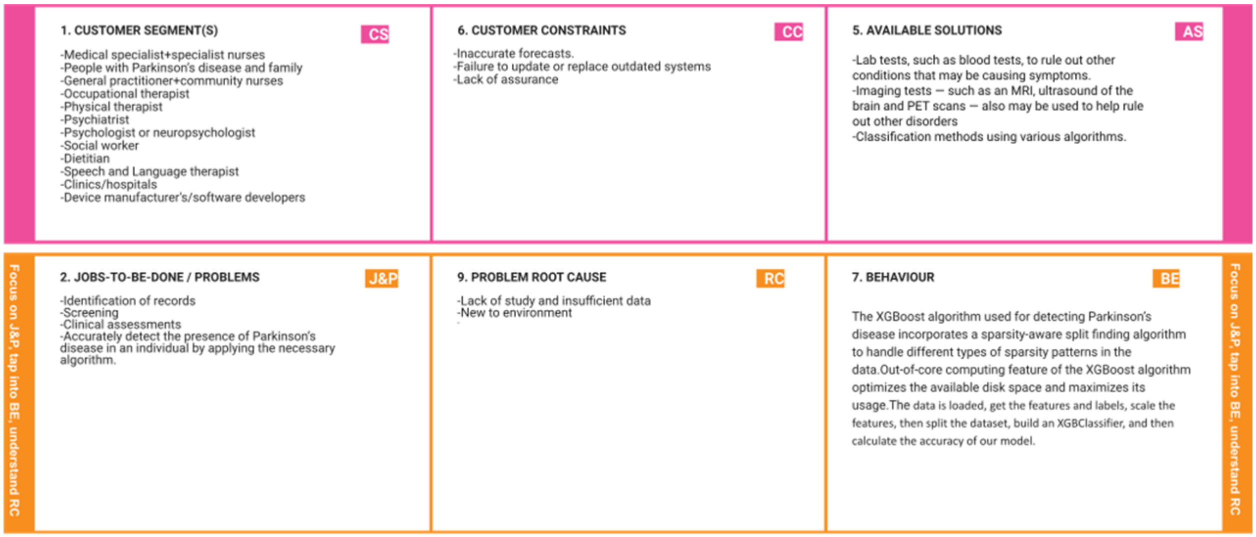

Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Algorithms & Models
Adaptive Boosting algorithm
Sensors
VOC Sensors
Timed Up and Go
Inertial sensors
Wearable Sensors
Data sets
Gait signals data set
Key stroke features
MRI data sets
ML Score Estimation
UDPRS score
ML methods
Neural networks-PD detection using nocturnal breathing
computer vision-PD detection using facial expressions
Detection of PD using wearable sensors, handwritten drawings, voice data
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Importance
Feasibility
Regardless of their importance, which tasks are more feasible than others?

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Parkinson's disease is caused by the disruption of brain cells that produce a substance to allow brain cells to communicate with each other called dopamine.It is a progressive disorder of the central nervous system affecting movement and inducing tremors and stiffness.The symptoms usually emerge slowly, and as the disease worsens, non-motor symptoms become more common. The most obvious early symptoms are tremor, rigidity, slowness of movement, and difficulty with walking.
2.	Idea / Solution description	The project aims at presenting a solution for Parkinson's disease detection using the Python libraries,scikit-learn, numpy, pandas, and xgboost. We'll load the data, get the features and labels, scale the features, then split the dataset, build an XGBClassifier, and then calculate the accuracy of our model. The main idea behind the implementation is to classify a person as Healthy or having Parkinson's disease by building a model using XGBoost.

3.	Novelty / Uniqueness	The XGBoost algorithm used for detecting Parkinson’s disease incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data. Out-of-core computing feature of the XGBoost algorithm optimizes the available disk space and maximizes its usage.
4.	Social Impact / Customer Satisfaction	Early diagnosis and treatment of PD are paramount to reducing the risk of disease progression, limiting the effects of PD on QoL, and potentially lowering long-term treatment costs.The proposed solution aims at predicting early Parkinson Disease in people using various factors.
5.	Business Model (Revenue Model)	Key partners: • Distributors • Academia • Platforms Key activities: Development of solutions Data acquisition Platform operation • Clinical Trials Key Resources: Data • People Value Propositions: Uniqueness Performance • Cost reduction Customer Segments: Clinics,Hospitals Software/platform developers
6.	Scalability of the Solution	XGBooster with different calculations the exactness, accuracy, review, and so forth is extremely excellent.XGBooster is not only able to keep up with all those other algorithms but exceeds them in performance.XGBoost can solve real-world scale problems using a minimal amount of resources.

3.4 PROBLEM SOLUTION FIT



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

1. CUSTOMER SEGMENT(S) CS

-Medical specialist+specialist nurses
-People with Parkinson's disease and family
-General practitioner+community nurses
-Occupational therapist
-Physical therapist
-Psychiatrist
-Psychologist or neuropsychologist
-Social worker
-Dietitian
-Speech and Language therapist
-Clinics/hospitals
-Device manufacturer's/software developers

6. CUSTOMER CONSTRAINTS CC

-Inaccurate forecasts.
-Failure to update or replace outdated systems
-Lack of assurance

5. AVAILABLE SOLUTIONS AS

-Lab tests, such as blood tests, to rule out other conditions that may be causing symptoms.
-Imaging tests — such as an MRI, ultrasound of the brain and PET scans — also may be used to help rule out other disorders
-Classification methods using various algorithms.

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

-Identification of records
-Screening
-Clinical assessments
-Accurately detect the presence of Parkinson's disease in an individual by applying the necessary algorithm.

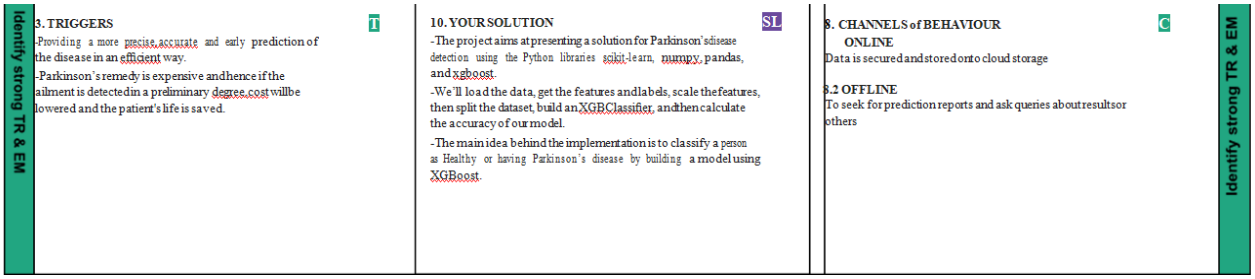
9. PROBLEM ROOT CAUSE RC

-Lack of study and insufficient data
-New to environment

7. BEHAVIOUR BE

The XGBoost algorithm used for detecting Parkinson's disease incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data.Out-of-core computing feature of the XGBoost algorithm optimizes the available disk space and maximizes its usage.The data is loaded, get the features and labels, scale the features, then split the dataset, build an XGBClassifier, and then calculate the accuracy of our model.

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



Identify strong TR & EM

3. TRIGGERS T

-Providing a more precise,accurate and early prediction of the disease in an efficient way.

-Parkinson's remedy is expensive and hence if the ailment is detected in a preliminary degree,cost will be lowered and the patient's life is saved.

10. YOUR SOLUTION SL

-The project aims at presenting a solution for Parkinson's disease detection using the Python libraries scikit-learn, numpy, pandas, and xgboost.

-We'll load the data, get the features and labels, scale the features, then split the dataset, build an XGBClassifier, and then calculate the accuracy of our model.

-The main idea behind the implementation is to classify a person as Healthy or having Parkinson's disease by building a model using XGBoost.

8. CHANNELS of BEHAVIOUR C

ONLINE

Data is secured and stored onto cloud storage

8.2 OFFLINE

To seek for prediction reports and ask queries about results or others

Identify strong TR & EM

E M	**4. EMOTIONS: BEFORE / AFTER** Before: -Uncertain about outcome of prediction -Drained emotionally and physically After: -Feeling relaxed and at ease after the prediction results and its accuracy -Sure and certain about the prediction and to take necessary medications.			

CHAPTER 4
REQUIREMENTS ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Authentication	Registration through Gmail, Login to the application, Confirmation via mail and OTP
FR-2	Data management	Web server has access to change/edit data and update to the server.
FR-3	Input data upload	Data is uploaded for analysis and prediction
FR-4	Testing	Applying the algorithms on the test data
FR-5	Prediction	Prediction is made by the model
FR-5	Result	Results of presence of Parkinson or not is displayed

4.2 NON FUNCTIONAL REQUIREMENTS

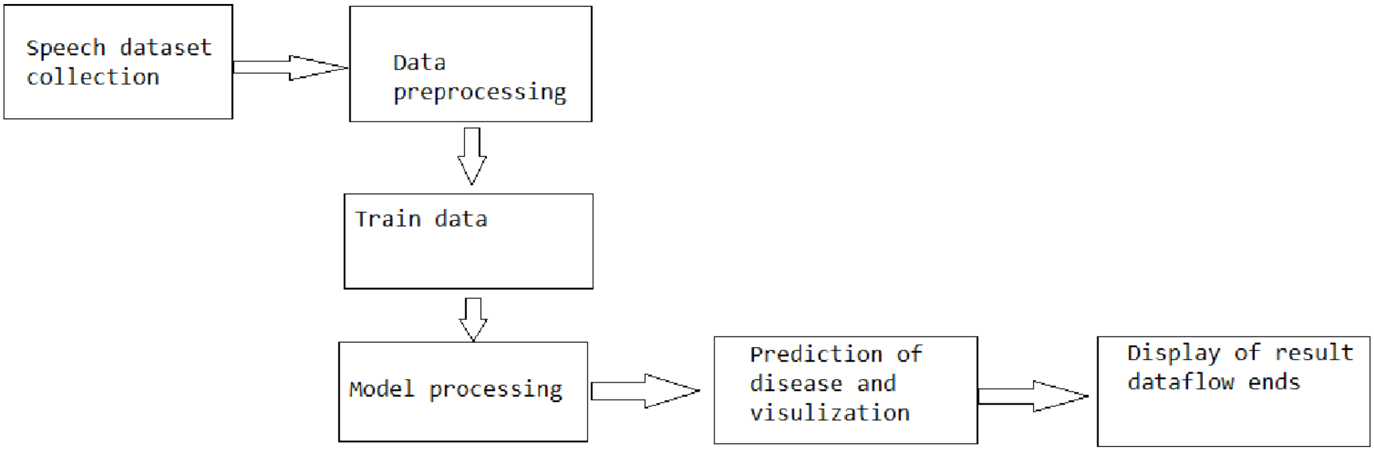
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The UI of the application must be user-friendly and easy to use. The input loading should be enabled faster.
NFR-2	**Security**	The image and voice records should be secure and must not be accessible by everyone.
NFR-3	**Reliability**	The prediction of the system must be with higher accuracy so that it will be trusted by users.

NFR-4	**Performance**	The XGBoost algorithm used for detecting PD should incorporate a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data. Out-of-core computing feature of the XGBoost algorithm should optimize the available disk space and maximizes its usage.
NFR-5	**Availability**	The application should be available to all groups of people all the time.
NFR-6	**Scalability**	XGBooster should not only be able to keep up with all those other algorithms but exceed them in performance. XGBoost should be able to solve real- world scale problems using a minimal number of resources.

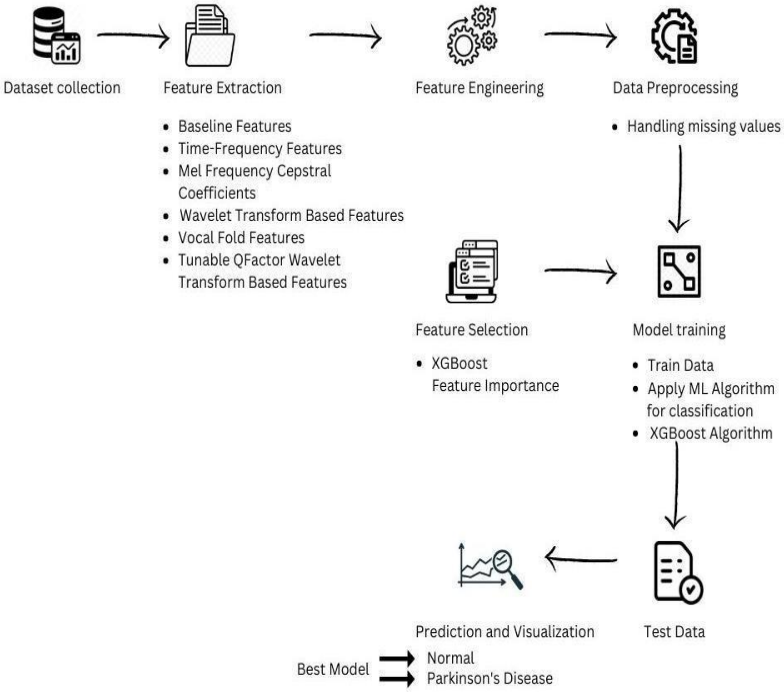
CHAPTER 5

PROJECT DESIGN

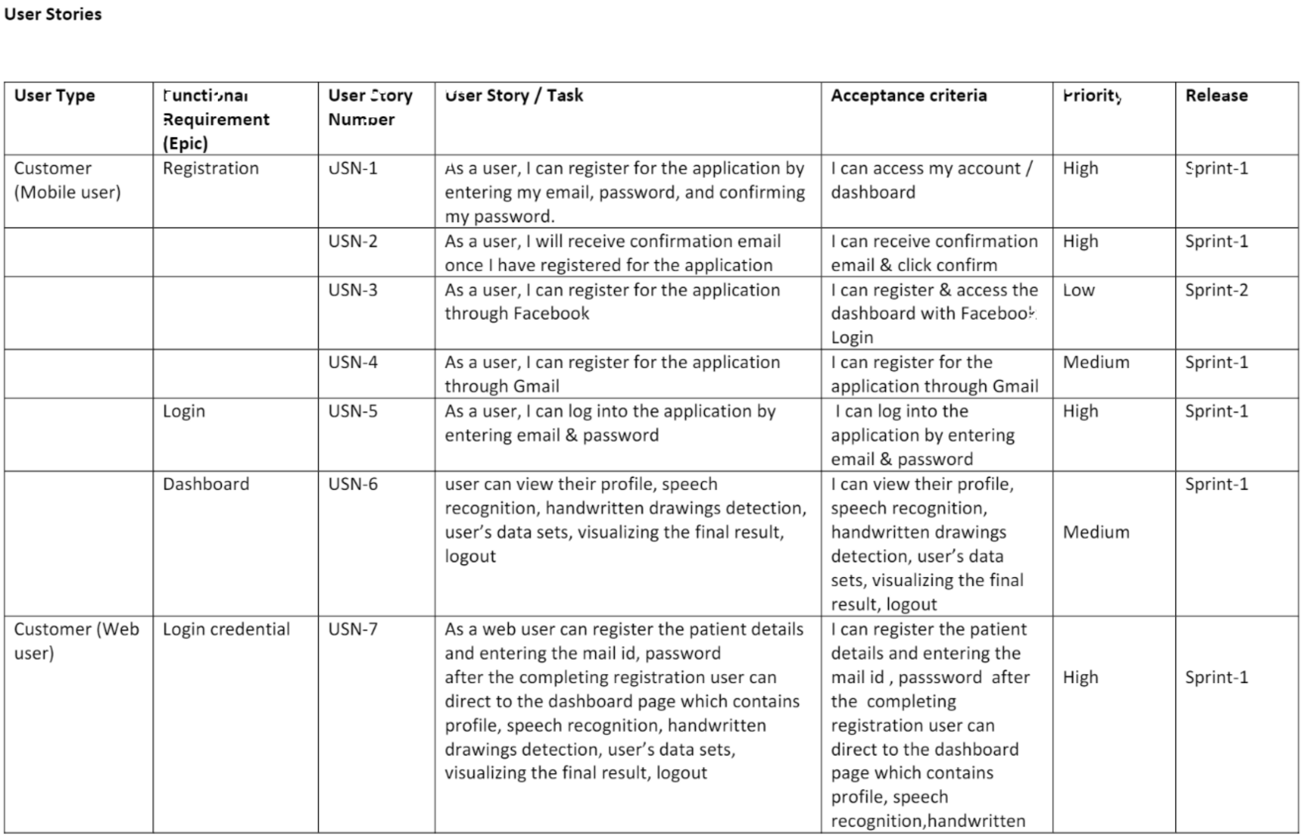
5.1 DATA FLOW DIAGRAM


Speech dataset collection
Data preprocessing
Train data
Model processing
Prediction of disease and visulization
Display of result dataflow ends

5.2 SOLUTION & TECHNICAL ARCHITECTURE

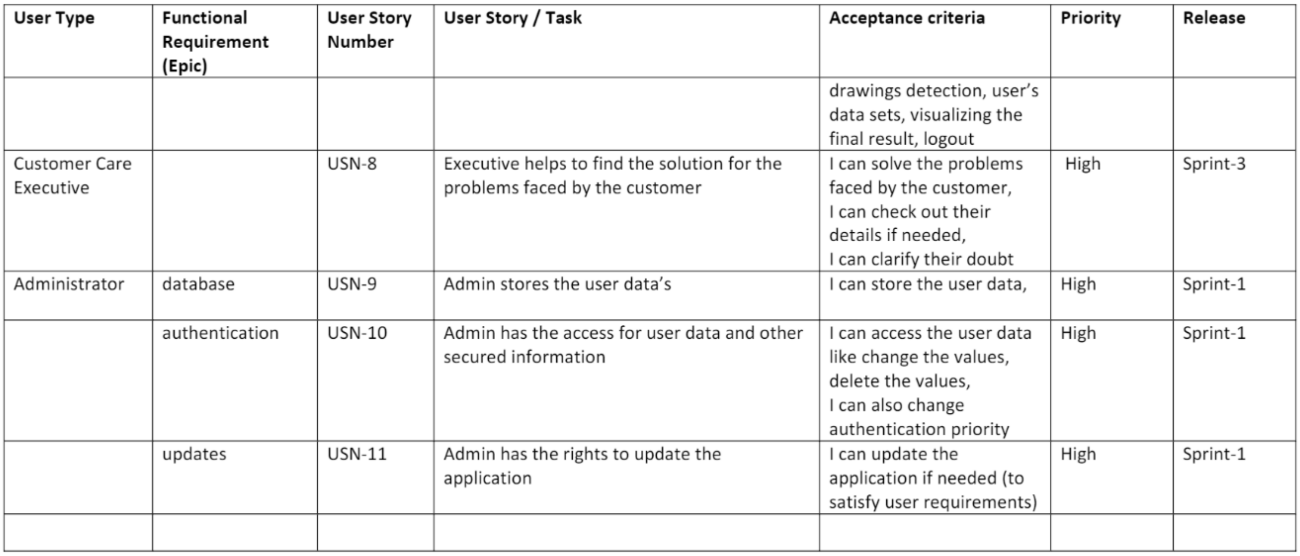

Dataset collection
Feature Extraction
• Baseline Features
• Time-Frequency Features
• Mel Frequency Cepstral Coefficients
• Wavelet Transform Based Features
• Vocal Fold Features
• Tunable QFactor Wavelet Transform Based Features
Feature Engineering
Data Preprocessing
• Handling missing values
Feature Selection
• XGBoost Feature Importance
Model training
• Train Data
• Apply ML Algorithm for classification
• XGBoost Algorithm
Prediction and Visualization
Test Data
Best Model
Normal
Parkinson's Disease

5.3 USER STORIES



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 confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register for the application through Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can log into the application by entering email & password	High	Sprint-1
	Dashboard	USN-6	user can view their profile, speech recognition, handwritten drawings detection, user's data sets, visualizing the final result, logout	I can view their profile, speech recognition, handwritten drawings detection, user's data sets, visualizing the final result, logout	Medium	Sprint-1
Customer (Web user)	Login credential	USN-7	As a web user can register the patient details and entering the mail id, password after the completing registration user can direct to the dashboard page which contains profile, speech recognition, handwritten drawings detection, user's data sets, visualizing the final result, logout	I can register the patient details and entering the mail id , passsword after the completing registration user can direct to the dashboard page which contains profile, speech recognition,handwritten	High	Sprint-1

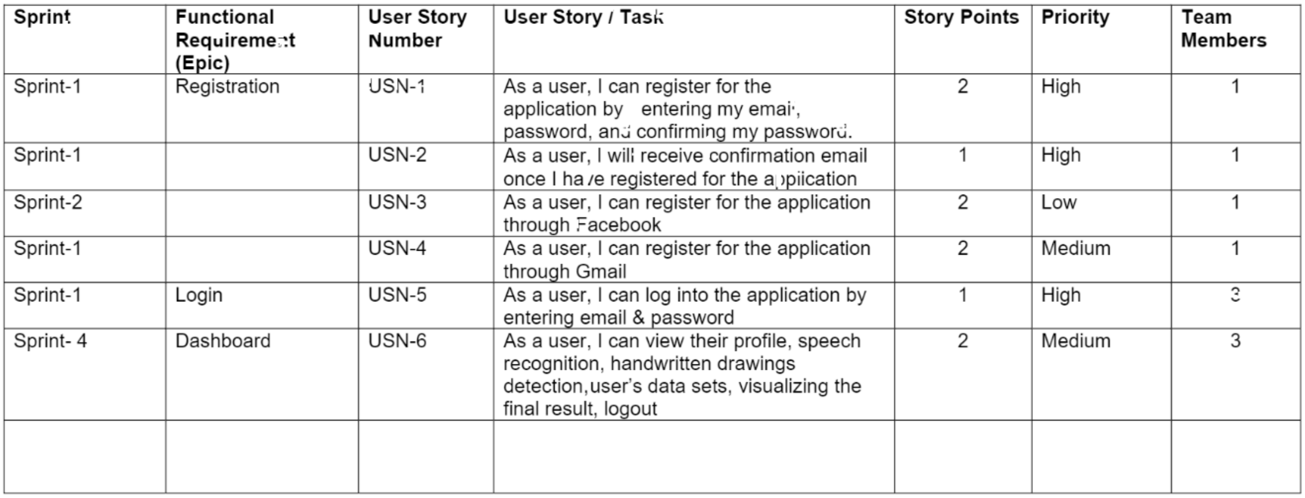


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
				drawings detection, user's data sets, visualizing the final result, logout		
Customer Care Executive		USN-8	Executive helps to find the solution for the problems faced by the customer	I can solve the problems faced by the customer, I can check out their details if needed, I can clarify their doubt	High	Sprint-3
Administrator	database	USN-9	Admin stores the user data's	I can store the user data,	High	Sprint-1
	authentication	USN-10	Admin has the access for user data and other secured information	I can access the user data like change the values, delete the values, I can also change authentication priority	High	Sprint-1
	updates	USN-11	Admin has the rights to update the application	I can update the application if needed (to satisfy user requirements)	High	Sprint-1

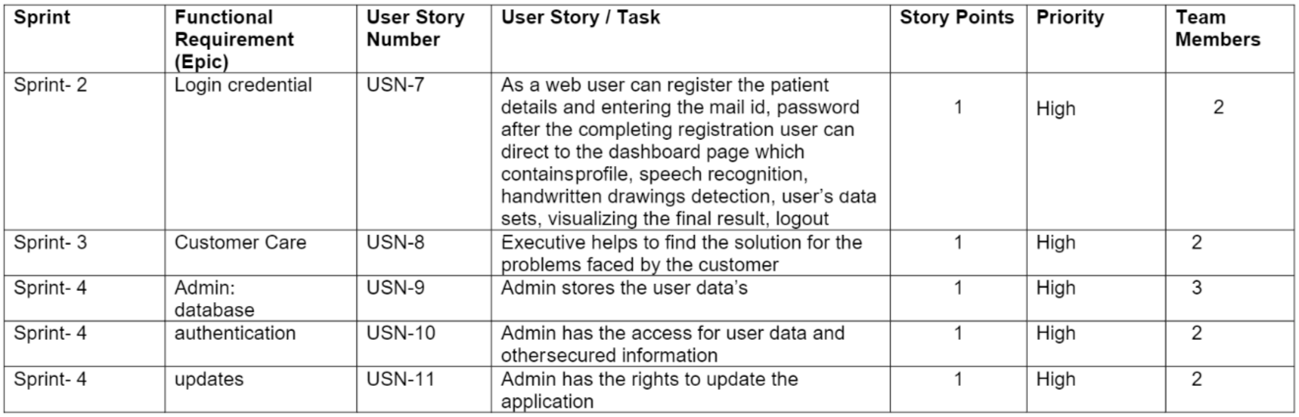
CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	1
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	1
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	1
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	1
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	3
Sprint- 4	Dashboard	USN-6	As a user, I can view their profile, speech recognition, handwritten drawings detection,user's data sets, visualizing the final result, logout	2	Medium	3

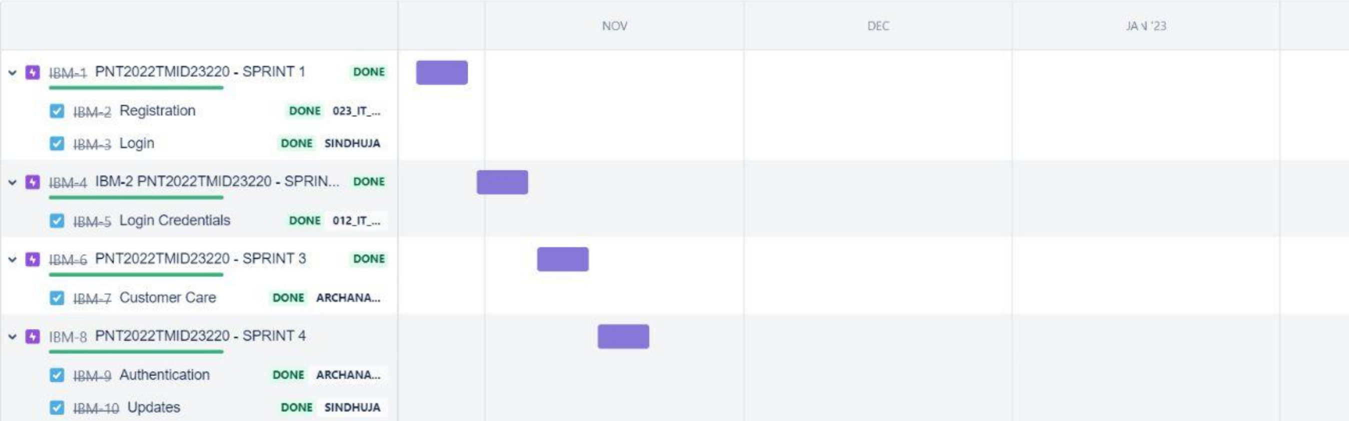


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint- 2	Login credential	USN-7	As a web user can register the patient details and entering the mail id, password after the completing registration user can direct to the dashboard page which containsprofile, speech recognition, handwritten drawings detection, user's data sets, visualizing the final result, logout	1	High	2
Sprint- 3	Customer Care	USN-8	Executive helps to find the solution for the problems faced by the customer	1	High	2
Sprint- 4	Admin: database	USN-9	Admin stores the user data's	1	High	3
Sprint- 4	authentication	USN-10	Admin has the access for user data and othersecured information	1	High	2
Sprint- 4	updates	USN-11	Admin has the rights to update the application	1	High	2

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

6.3 REPORTS FROM JIRA



~~IBM-1~~ PNT2022TMID23220 - SPRINT 1 DONE

☑ ~~IBM-2~~ Registration DONE 023_IT_...

☑ ~~IBM-3~~ Login DONE SINDHUJA

~~IBM-4~~ IBM-2 PNT2022TMID23220 - SPRIN... DONE

☑ ~~IBM-5~~ Login Credentials DONE 012_IT_...

~~IBM-6~~ PNT2022TMID23220 - SPRINT 3 DONE

☑ ~~IBM-7~~ Customer Care DONE ARCHANA...

IBM-8 PNT2022TMID23220 - SPRINT 4

☑ ~~IBM-9~~ Authentication DONE ARCHANA...

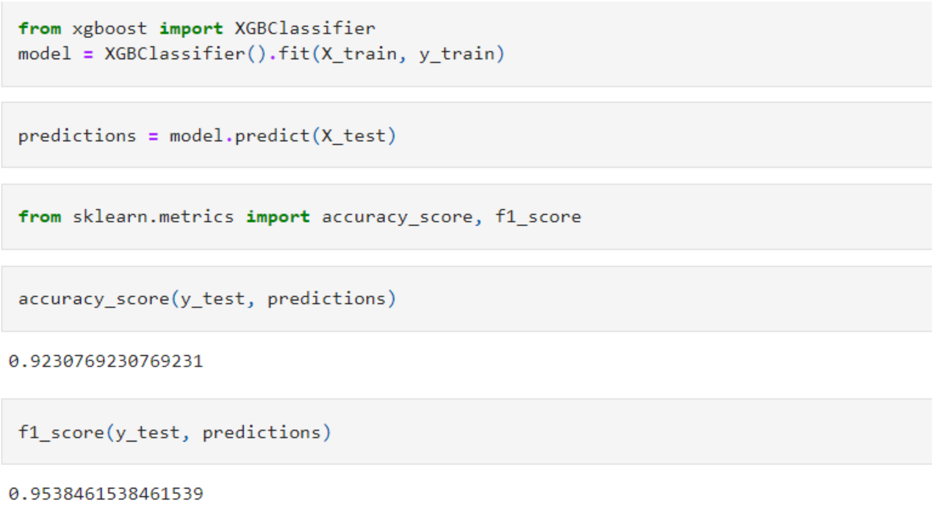
☑ ~~IBM-10~~ Updates DONE SINDHUJA

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1

The model has been trained to accurately predict Parkinson’s Disease using Machine learning.



```
from xgboost import XGBClassifier
model = XGBClassifier().fit(X_train, y_train)
```

```
predictions = model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, f1_score
```

```
accuracy_score(y_test, predictions)
```

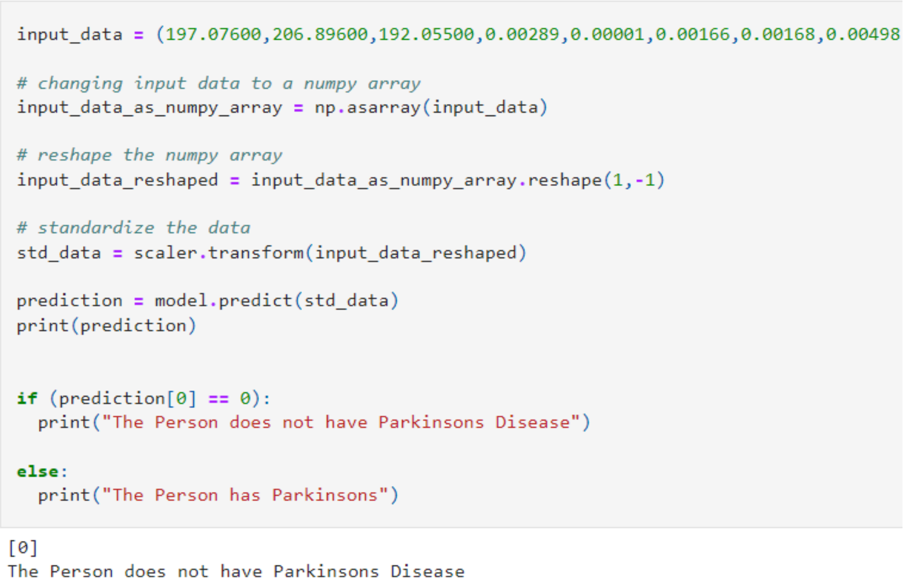
```
0.9230769230769231
```

```
f1_score(y_test, predictions)
```

```
0.9538461538461539
```

7.2 FEATURE 2

Early onset predictions are made if a person is affected by PD or not.



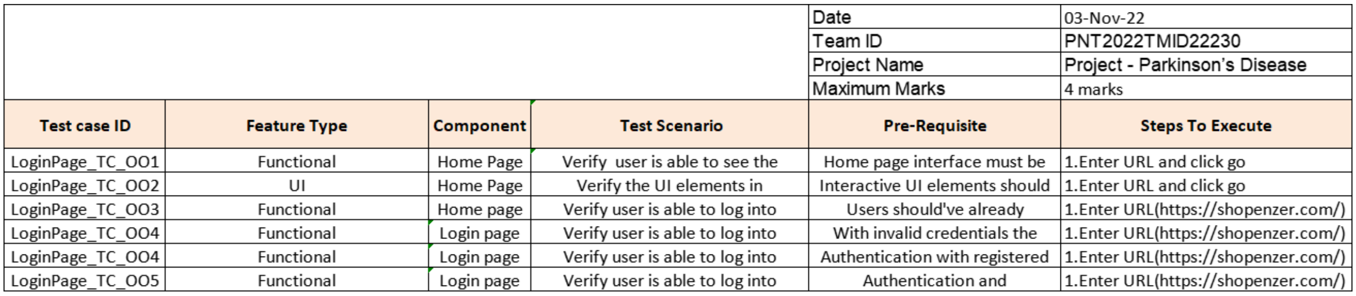
```
input_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498

# changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the data
std_data = scaler.transform(input_data_reshaped)

prediction = model.predict(std_data)
print(prediction)


if (prediction[0] == 0):
  print("The Person does not have Parkinsons Disease")

else:
  print("The Person has Parkinsons")
```

```
[0]
The Person does not have Parkinsons Disease
```

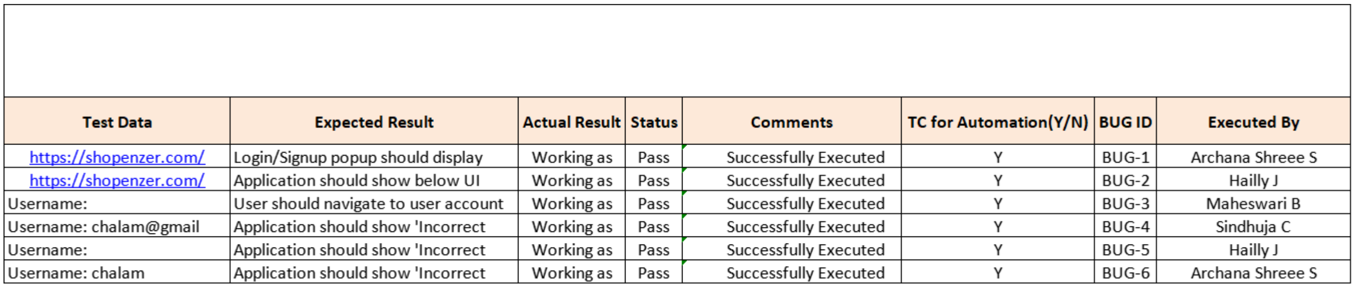
CHAPTER 8

TESTING

8.1 TEST CASES

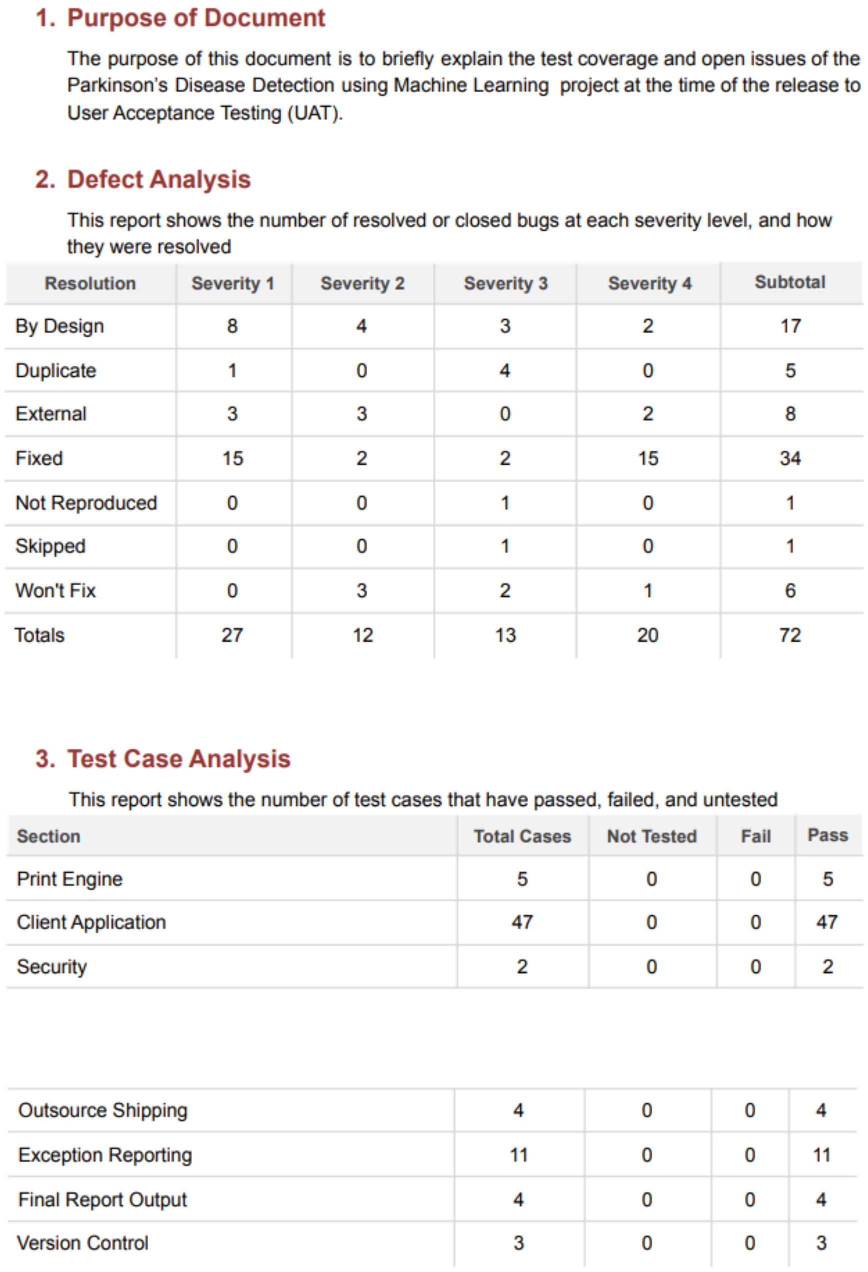


				Date	03-Nov-22
				Team ID	PNT2022TMID22230
				Project Name	Project - Parkinson's Disease
				Maximum Marks	4 marks
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the	Home page interface must be	1.Enter URL and click go
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in	Interactive UI elements should	1.Enter URL and click go
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into	Users should've already	1.Enter URL(https://shopenzer.com/)
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into	With invalid credentials the	1.Enter URL(https://shopenzer.com/)
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into	Authentication with registered	1.Enter URL(https://shopenzer.com/)
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into	Authentication and	1.Enter URL(https://shopenzer.com/)



Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
https://shopenzer.com/	Login/Signup popup should display	Working as	Pass	Successfully Executed	Y	BUG-1	Archana Shreee S
https://shopenzer.com/	Application should show below UI	Working as	Pass	Successfully Executed	Y	BUG-2	Hailly J
Username:	User should navigate to user account	Working as	Pass	Successfully Executed	Y	BUG-3	Maheswari B
Username: chalam@gmail	Application should show 'Incorrect	Working as	Pass	Successfully Executed	Y	BUG-4	Sindhuja C
Username:	Application should show 'Incorrect	Working as	Pass	Successfully Executed	Y	BUG-5	Hailly J
Username: chalam	Application should show 'Incorrect	Working as	Pass	Successfully Executed	Y	BUG-6	Archana Shreee S

8.2 USER ACCEPTANCE TESTING



1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	3	2	17
Duplicate	1	0	4	0	5
External	3	3	0	2	8
Fixed	15	2	2	15	34
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	0	1
Won't Fix	0	3	2	1	6
Totals	27	12	13	20	72

3. Test Case Analysis

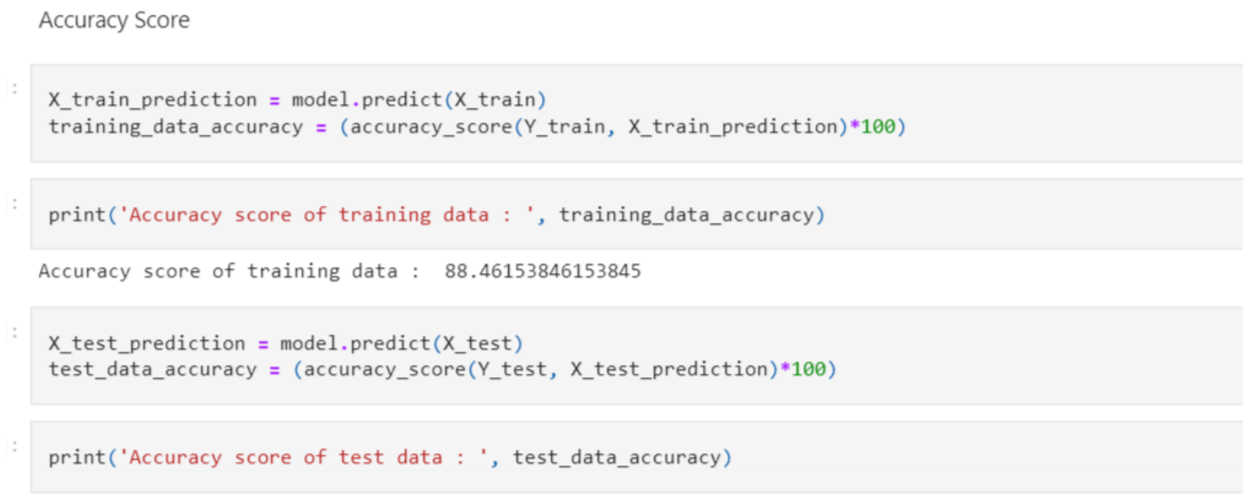
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	47	0	0	47
Security	2	0	0	2
Outsource Shipping	4	0	0	4
Exception Reporting	11	0	0	11
Final Report Output	4	0	0	4
Version Control	3	0	0	3

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS



Accuracy Score

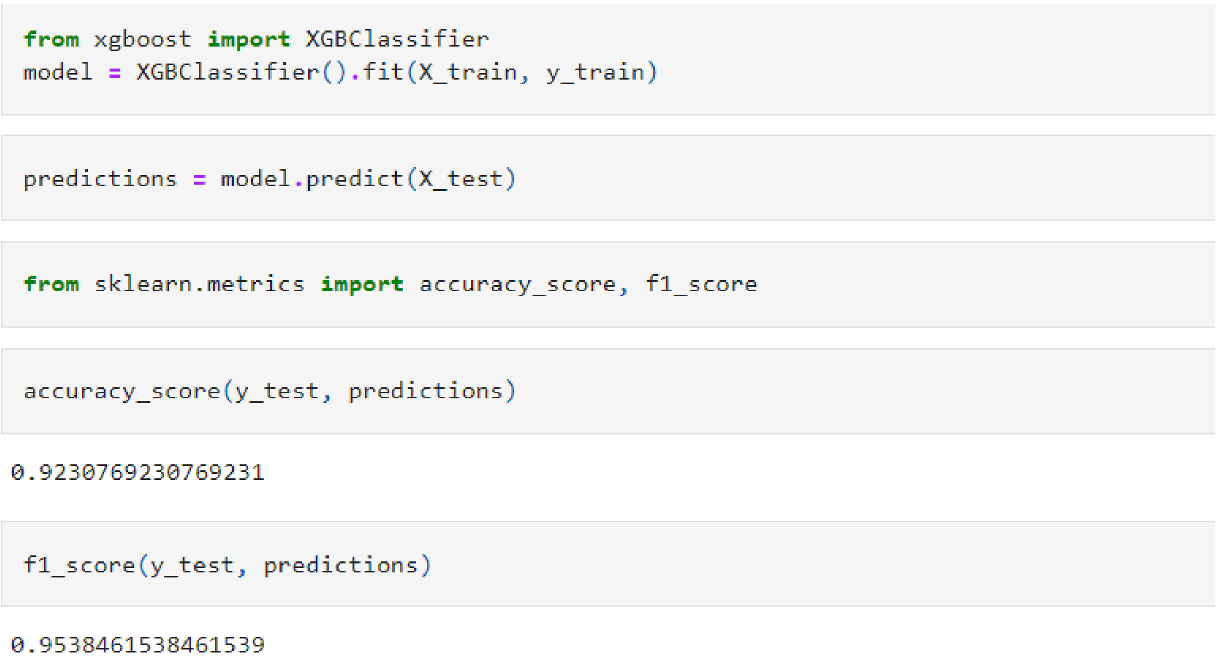
```
X_train_prediction = model.predict(X_train)
training_data_accuracy = (accuracy_score(Y_train, X_train_prediction)*100)
```

```
print('Accuracy score of training data : ', training_data_accuracy)
```

```
Accuracy score of training data :  88.46153846153845
```

```
X_test_prediction = model.predict(X_test)
test_data_accuracy = (accuracy_score(Y_test, X_test_prediction)*100)
```

```
print('Accuracy score of test data : ', test_data_accuracy)
```



```
from xgboost import XGBClassifier
model = XGBClassifier().fit(X_train, y_train)
```

```
predictions = model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, f1_score
```

```
accuracy_score(y_test, predictions)
```

```
0.9230769230769231
```

```
f1_score(y_test, predictions)
```

```
0.9538461538461539
```

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Less time consuming
- More accuracy in the model
- Easily implemented
- Could assess a variety of factors
- Certainty of diagnosis

DISADVANTAGES

- Packages to be installed
- Data collection is difficult
- May have long lead time
- Low conversion rates
- Non manifesting participants hard to find

CHAPTER 11

CONCLUSION

The researchers found that the drawing speed was slower and the pen pressure is lower among Parkinson's patients. One of the indications of Parkinson's is tremors and rigidity in the muscles, making it difficult to draw smooth spirals and waves. It is possible to detect Parkinson's disease using the drawings alone instead of measuring the speed and pressure of the pen on paper. Our goal is to quantify the visual appearance(using HOG method) of these drawings and then train a machine learning model to classify them. In this project, We are using Histogram of Oriented Gradients (HOG) image descriptor along with a XGBoost Classifier to automatically detect Parkinson's disease in hand-drawn images of spirals and waves. Here, we presented included studies in a high-level summary, providing access to information including (a) machine learning methods that have been used in the diagnosis of PD and associated outcomes, (b) types of clinical, behavioral and biometric data that could be used for rendering more accurate diagnoses, (c) potential biomarkers for assisting clinical decision making, and (d) other highly relevant information, including databases that could be used to enlarge and enrich smaller datasets. In summary, realization of machine learning-assisted diagnosis of PD yields high potential for a more systematic clinical decision-making system, while adaptation of novel biomarkers may give rise to easier access to PD diagnosis at an earlier stage. Machine learning approaches therefore have the potential to provide clinicians with additional tools to screen, detect or diagnose PD.

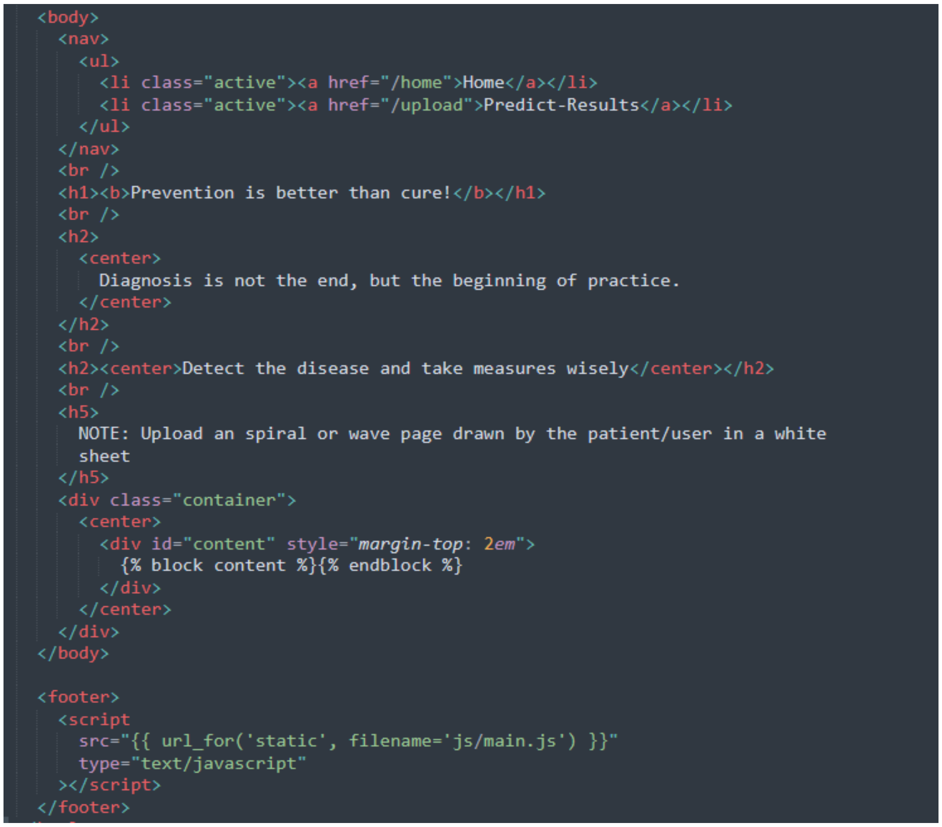
CHAPTER 12

FUTURE SCOPE

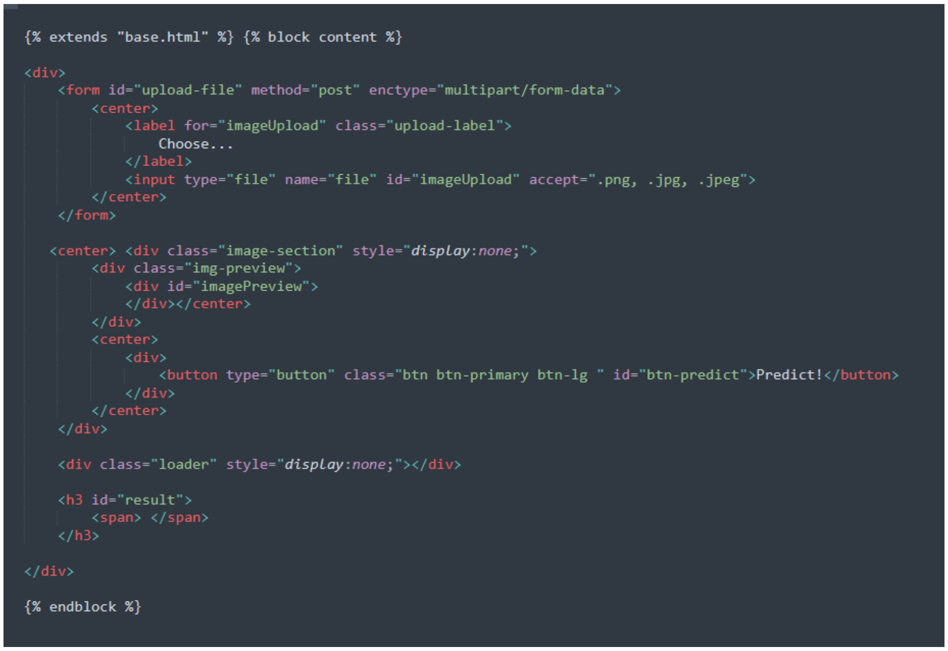
The model can be trained with an enormous amount of data to improve the accuracy. We can also merge the voice dataset and train the model accordingly for higher productivity.In future, these models can be trained with different datasets that have best features and can be predicted more accurately. If the accuracy rate increases, it can be used by the laboratories and hospitals so that it is easy to predict in early stages. These models can be also used with different medical and disease datasets. In future the work can be extended by building a hybrid model that can find more than one disease with an accurate dataset and that dataset has common features of two diseases. In future the work can be extended to build a model that may extract more important features among all features in the dataset so that it produces more accuracy.

APPENDIX

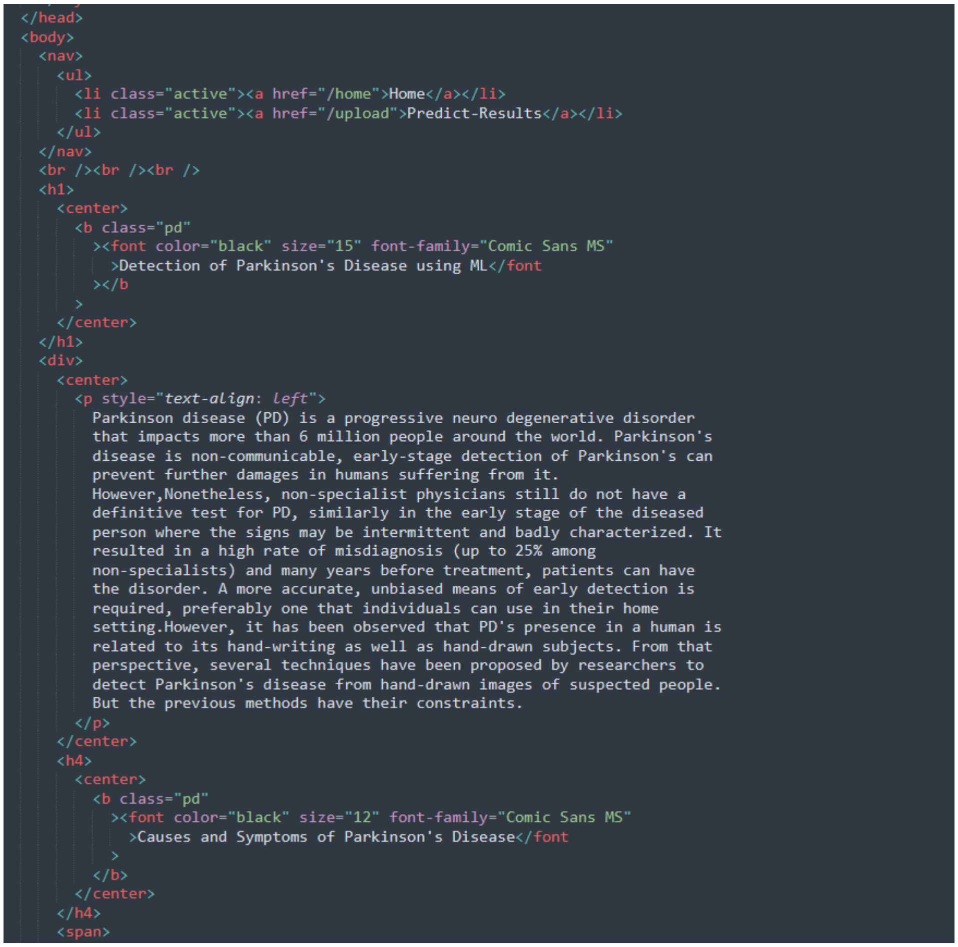
SOURCE CODE



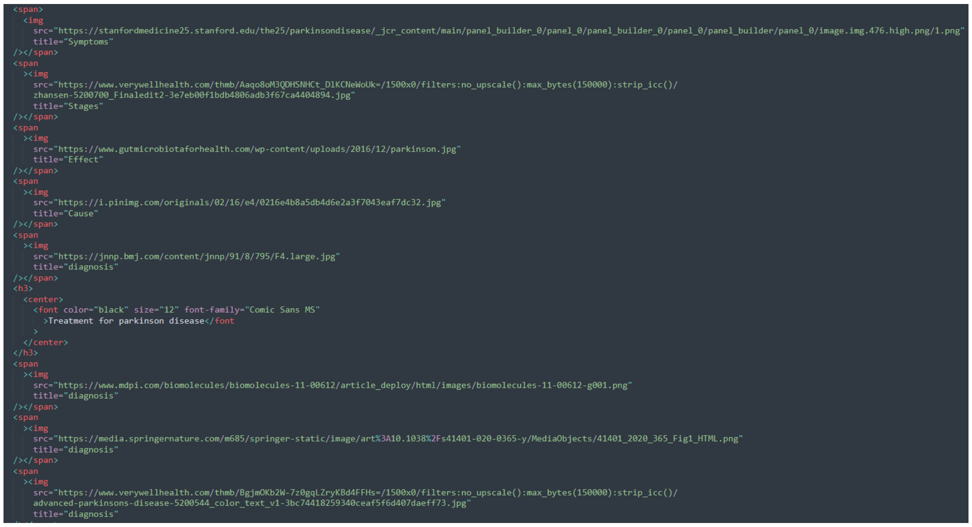
```
<body>
  <nav>
    <ul>
      <li class="active"><a href="/home">Home</a></li>
      <li class="active"><a href="/upload">Predict-Results</a></li>
    </ul>
  </nav>
  <br />
  <h1><b>Prevention is better than cure!</b></h1>
  <br />
  <h2>
    <center>
      Diagnosis is not the end, but the beginning 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>
```



```
{% 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 %}
```



```html
</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>
```



```html
<span>
  <img
    src="https://stanfordmedicine25.stanford.edu/the25/parkinsondisease/_jcr_content/main/panel_builder_0/panel_0/panel_builder_0/panel_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(150000):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"
```

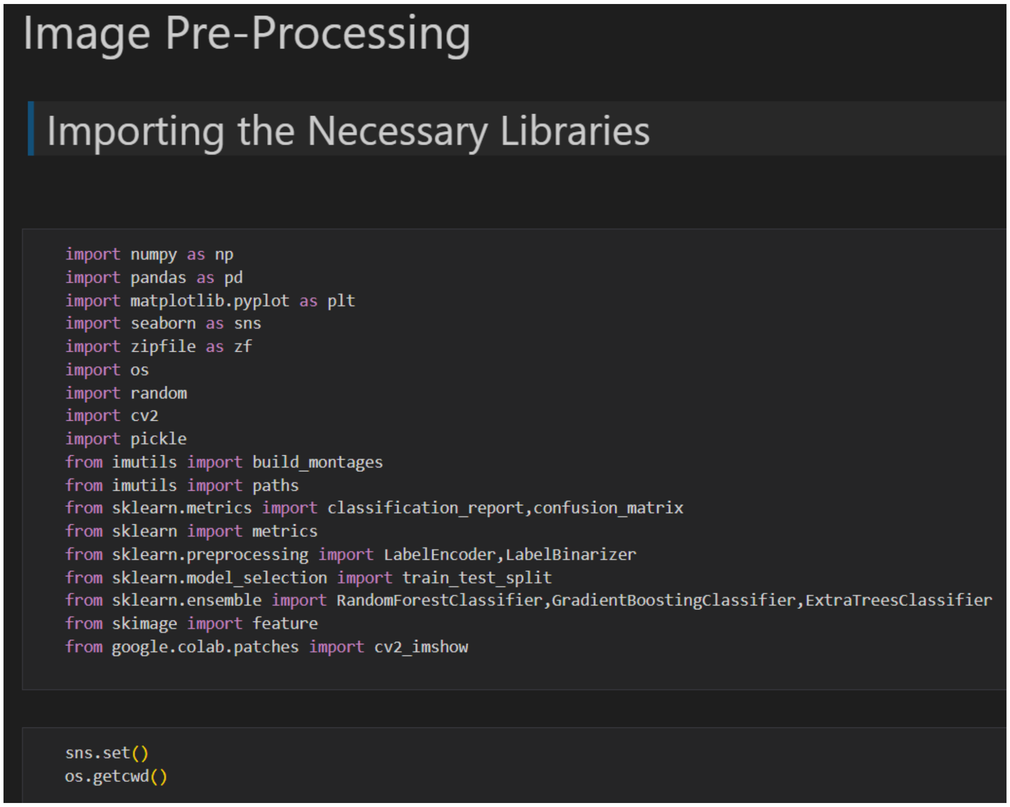
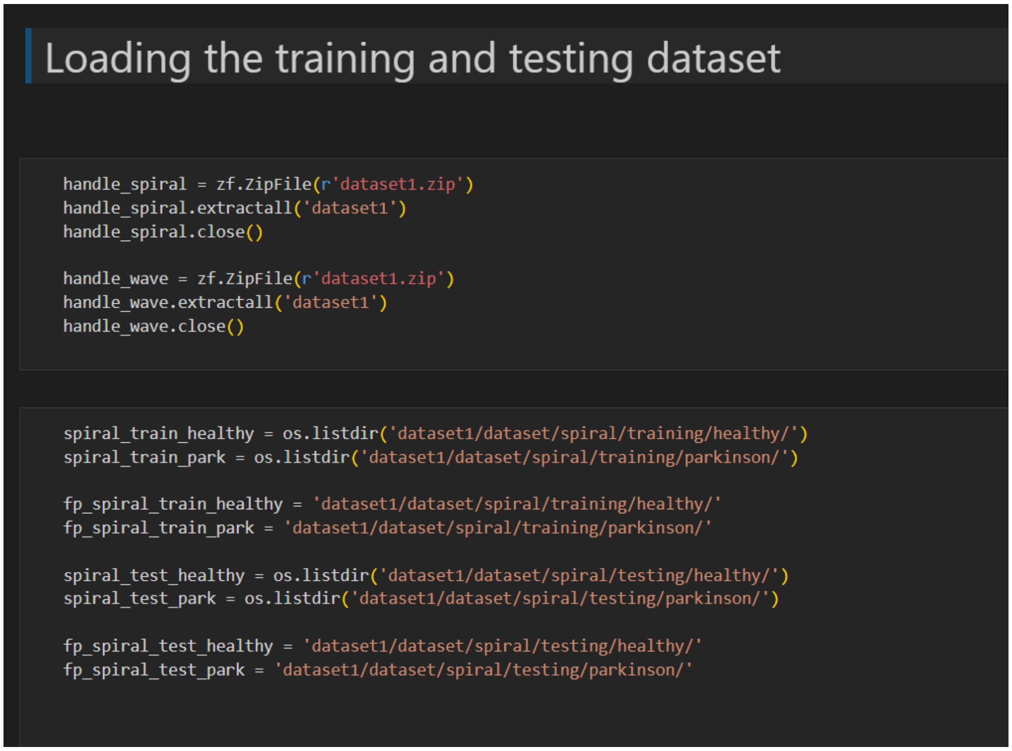


Image Pre-Processing

Importing the Necessary Libraries

```python
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
```

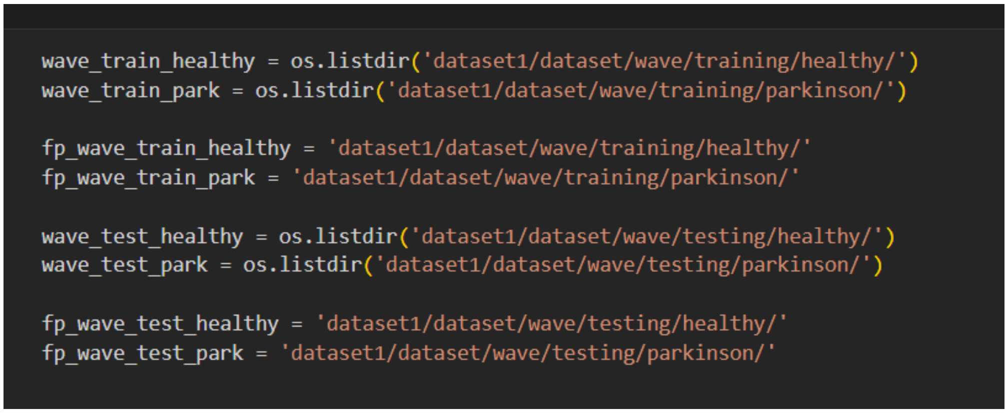
```python
sns.set()
os.getcwd()
```



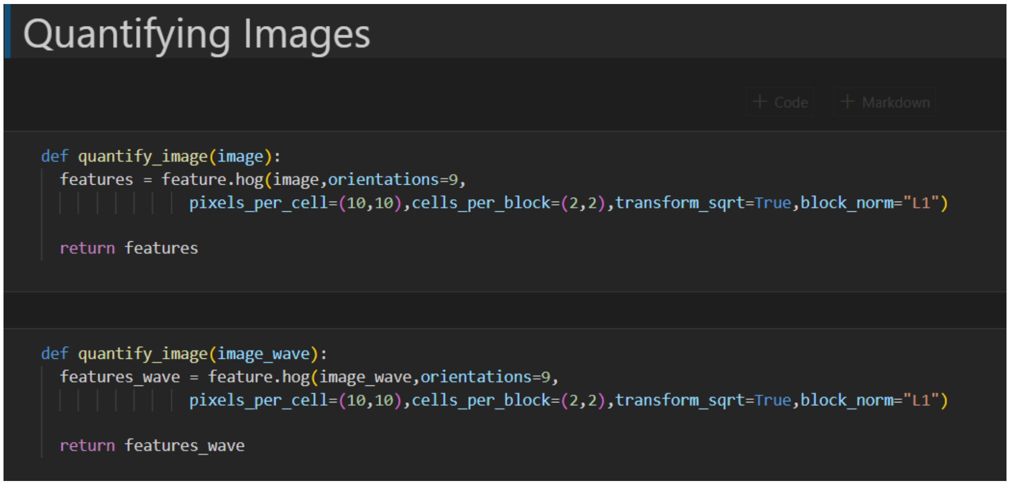
Loading the training and testing dataset

```python
handle_spiral = zf.ZipFile(r'dataset1.zip')
handle_spiral.extractall('dataset1')
handle_spiral.close()

handle_wave = zf.ZipFile(r'dataset1.zip')
handle_wave.extractall('dataset1')
handle_wave.close()
```

```python
spiral_train_healthy = os.listdir('dataset1/dataset/spiral/training/healthy/')
spiral_train_park = os.listdir('dataset1/dataset/spiral/training/parkinson/')

fp_spiral_train_healthy = 'dataset1/dataset/spiral/training/healthy/'
fp_spiral_train_park = 'dataset1/dataset/spiral/training/parkinson/'

spiral_test_healthy = os.listdir('dataset1/dataset/spiral/testing/healthy/')
spiral_test_park = os.listdir('dataset1/dataset/spiral/testing/parkinson/')

fp_spiral_test_healthy = 'dataset1/dataset/spiral/testing/healthy/'
fp_spiral_test_park = 'dataset1/dataset/spiral/testing/parkinson/'
```



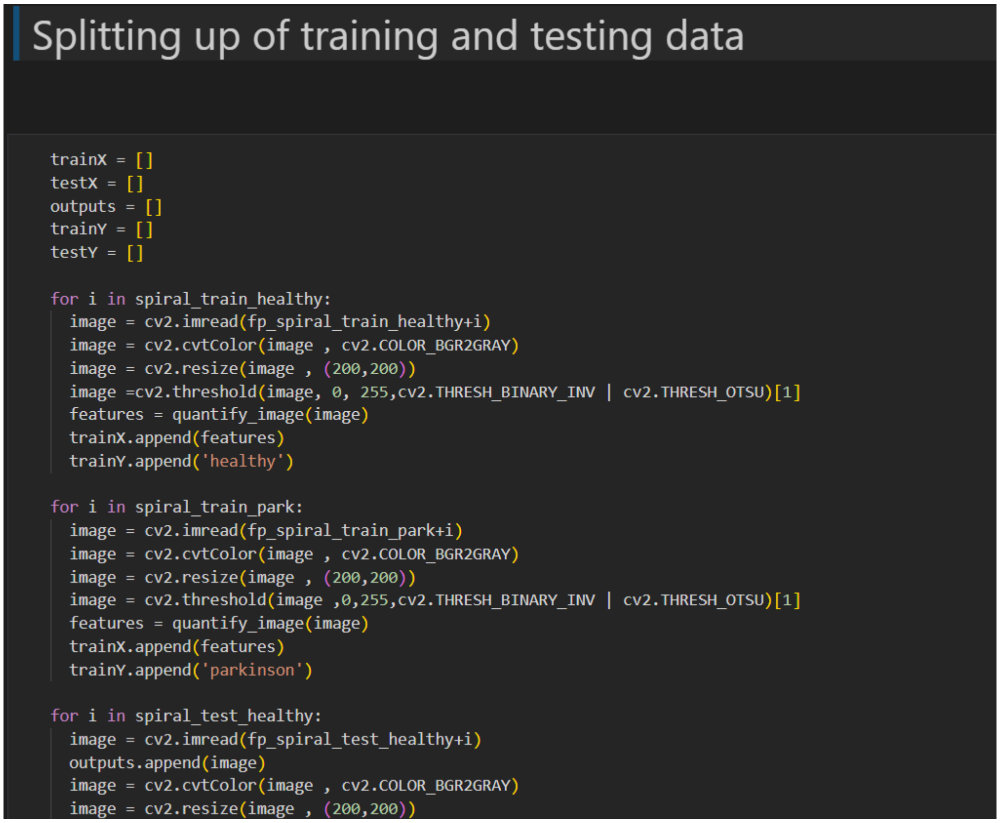
```
wave_train_healthy = os.listdir('dataset1/dataset/wave/training/healthy/')
wave_train_park = os.listdir('dataset1/dataset/wave/training/parkinson/')

fp_wave_train_healthy = 'dataset1/dataset/wave/training/healthy/'
fp_wave_train_park = 'dataset1/dataset/wave/training/parkinson/'

wave_test_healthy = os.listdir('dataset1/dataset/wave/testing/healthy/')
wave_test_park = os.listdir('dataset1/dataset/wave/testing/parkinson/')

fp_wave_test_healthy = 'dataset1/dataset/wave/testing/healthy/'
fp_wave_test_park = 'dataset1/dataset/wave/testing/parkinson/'
```



Quantifying Images

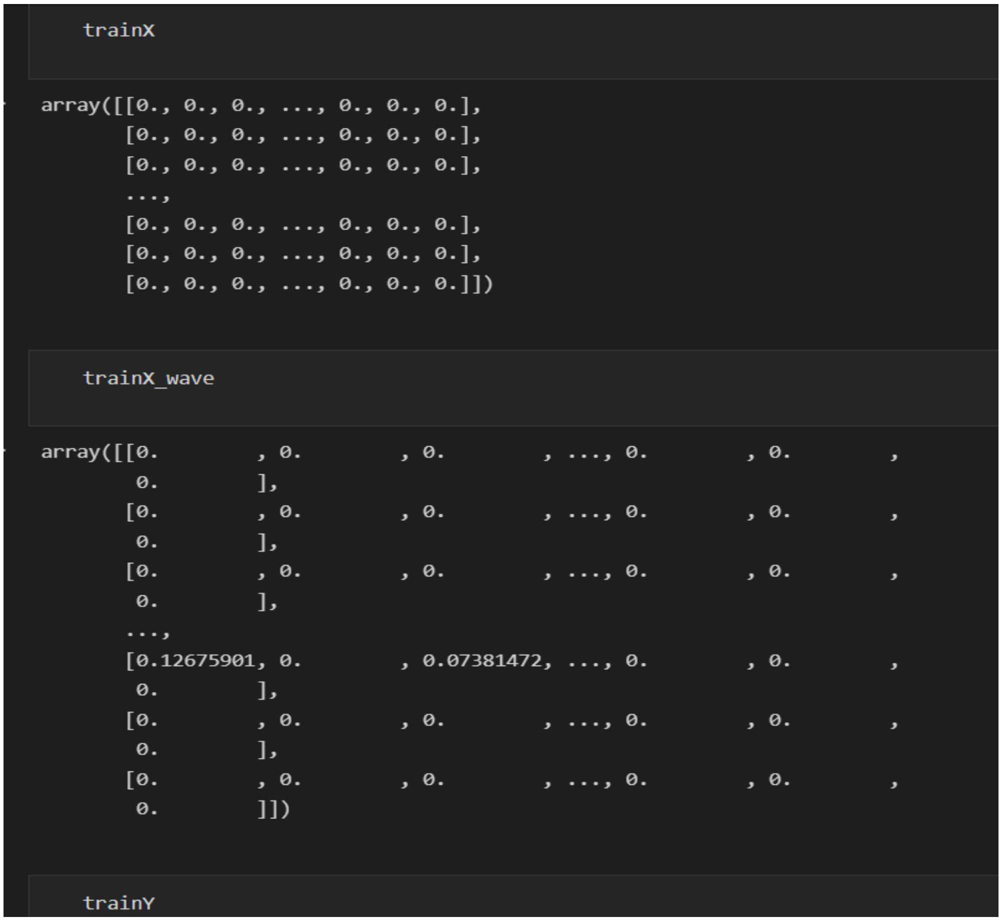
```
def quantify_image(image):
  features = feature.hog(image,orientations=9,
              pixels_per_cell=(10,10),cells_per_block=(2,2),transform_sqrt=True,block_norm="L1")

  return features
```

```
def quantify_image(image_wave):
  features_wave = feature.hog(image_wave,orientations=9,
              pixels_per_cell=(10,10),cells_per_block=(2,2),transform_sqrt=True,block_norm="L1")

  return features_wave
```



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('healthy')

for i in spiral_train_park:
  image = cv2.imread(fp_spiral_train_park+i)
  image = cv2.cvtColor(image , cv2.COLOR_BGR2GRAY)
  image = cv2.resize(image , (200,200))
  image = cv2.threshold(image ,0,255,cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
  features = quantify_image(image)
  trainX.append(features)
  trainY.append('parkinson')

for i in spiral_test_healthy:
  image = cv2.imread(fp_spiral_test_healthy+i)
  outputs.append(image)
  image = cv2.cvtColor(image , cv2.COLOR_BGR2GRAY)
  image = cv2.resize(image , (200,200))
```



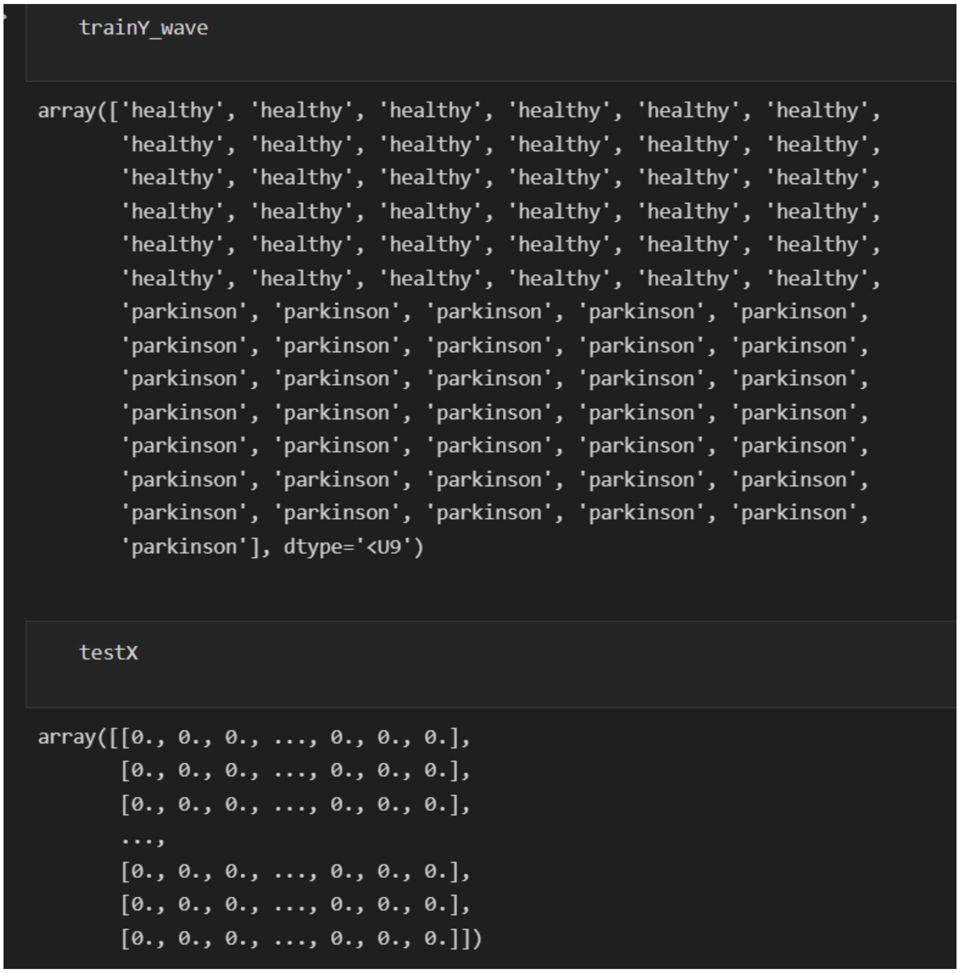
```
trainX
```

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

```
trainX_wave
```

```
array([[0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.12675901, 0.        , 0.07381472, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ]])
```

```
trainY
```

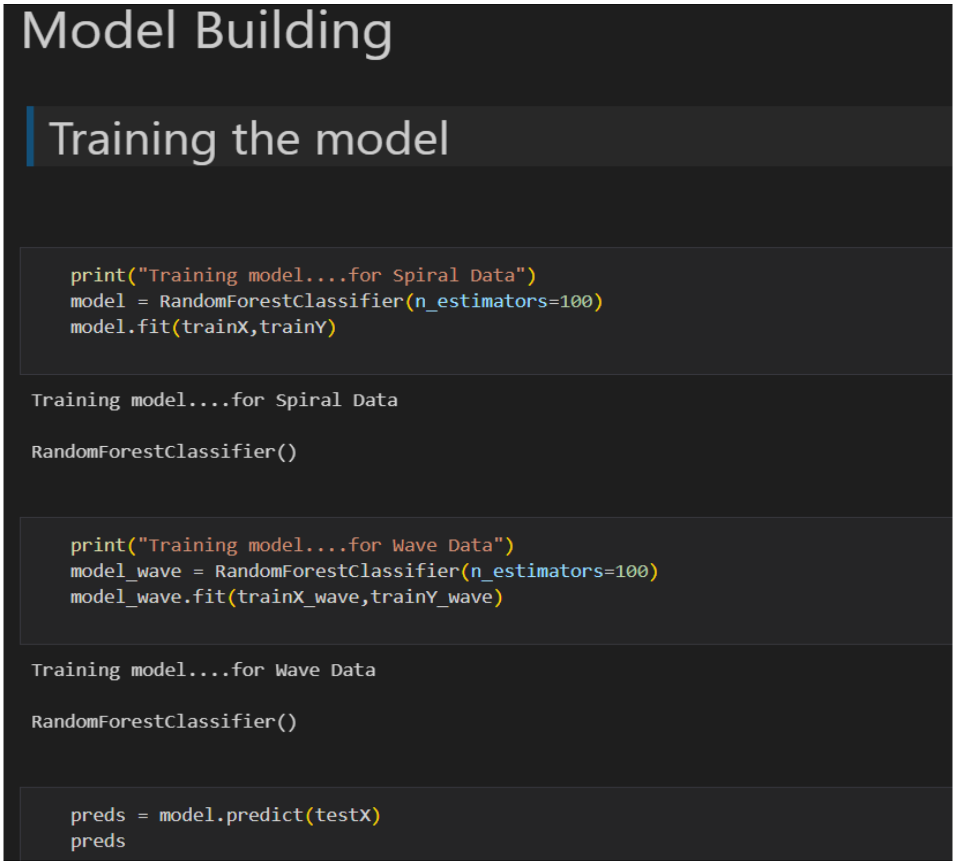


```
trainY_wave
```

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

```
testX
```

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



Model Building

Training the model

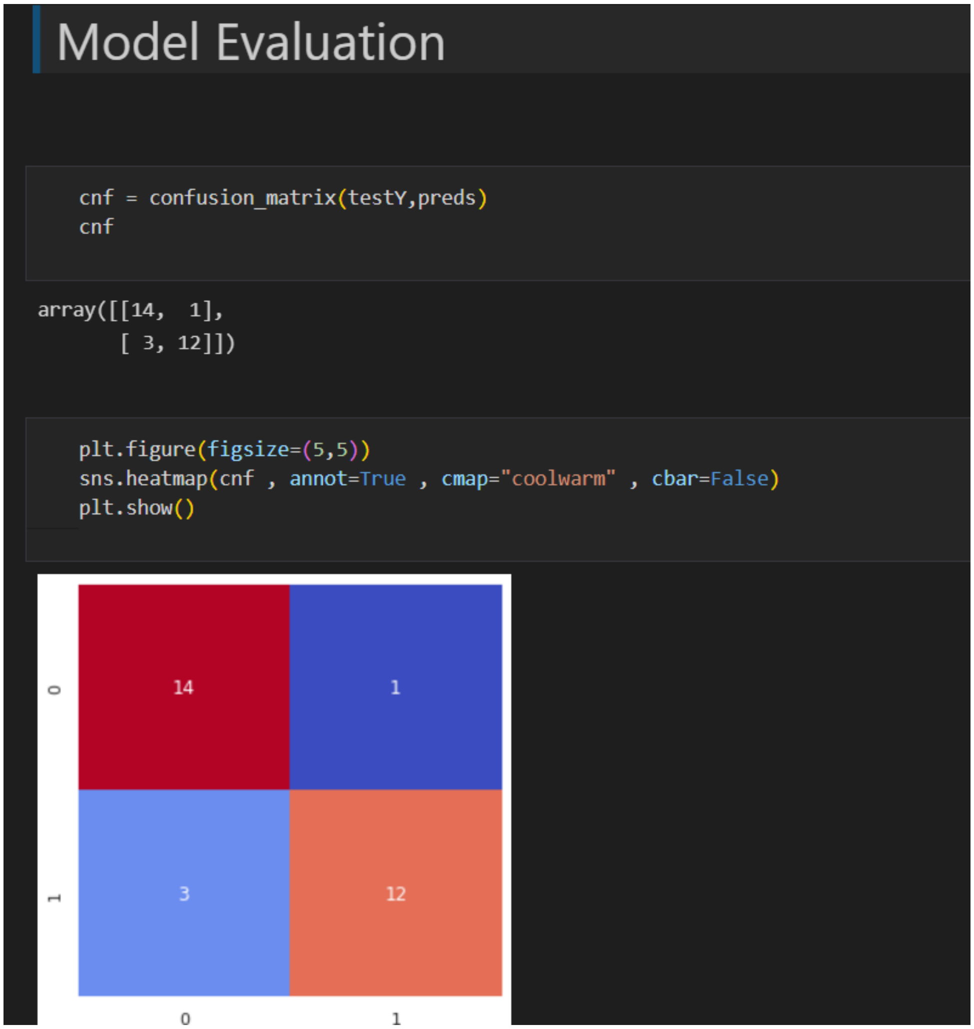
```
print("Training model....for Spiral Data")
model = RandomForestClassifier(n_estimators=100)
model.fit(trainX,trainY)
```

```
Training model....for Spiral Data

RandomForestClassifier()
```

```
print("Training model....for Wave Data")
model_wave = RandomForestClassifier(n_estimators=100)
model_wave.fit(trainX_wave,trainY_wave)
```

```
Training model....for Wave Data

RandomForestClassifier()
```

```
preds = model.predict(testX)
preds
```

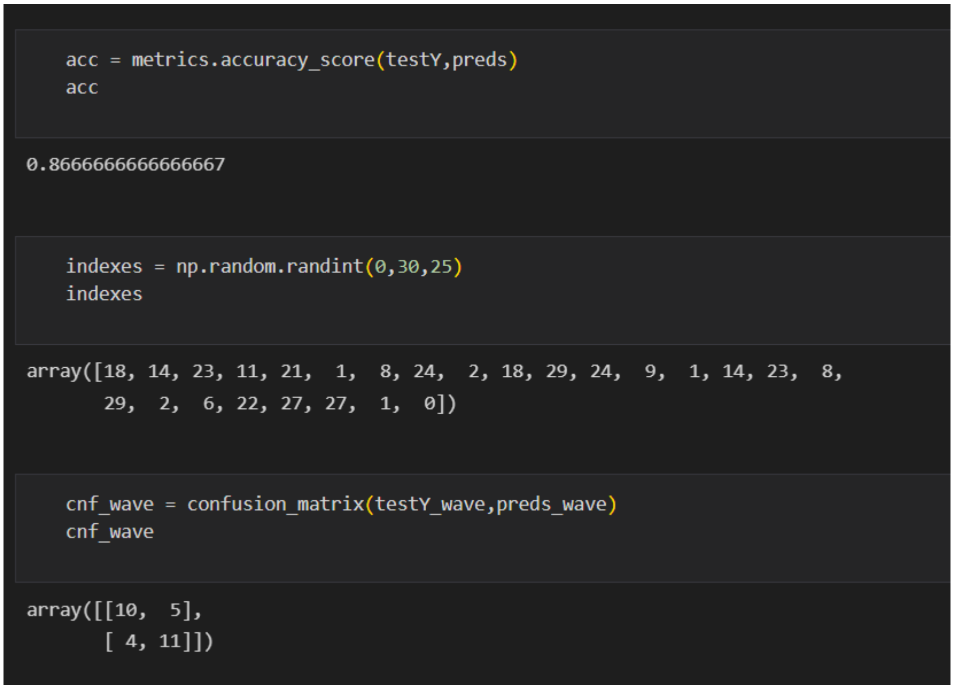


Model Evaluation

```
cnf = confusion_matrix(testY,preds)
cnf
```

```
array([[14,  1],
       [ 3, 12]])
```

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



```
acc = metrics.accuracy_score(testY,preds)
acc
```

```
0.8666666666666667
```

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

```
array([18, 14, 23, 11, 21,  1,  8, 24,  2, 18, 29, 24,  9,  1, 14, 23,  8,
       29,  2,  6, 22, 27, 27,  1,  0])
```

```
cnf_wave = confusion_matrix(testY_wave,preds_wave)
cnf_wave
```

```
array([[10,  5],
       [ 4, 11]])
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-36672-1660297040

PROJECT DEMO LINK:

https://drive.google.com/file/d/1drxtE5iwmshEqHJHZ-B4hi0swDfXJVhy/view?usp=drivesdk