DETECTION OF PARKINSON'S DISEASE USING MACHINE LEARNING

Project Report

Submitted by

THILLAIVALAVAN A S - 513419104049

RAGHUL J - 513419104035

SHERINE SHINY S - 513419104040

SALINI S - 513419104301

1. INTRODUCTION

1.1 Project Overview

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

1.2 Purpose

The purpose of the project is to detect Parkinson's disease in early stage by developing a user friendly application deployed with a machine learning model that detects the disease so that proper medication can be done to improve symptoms and quality of life of the individual.The detection is done by uploading a hand-drawn image of a spiral in which Histogram of Gradients(HOG) descriptor is applied and features are extracted which is further classified by Random Forest Classifier to give the exact output

2. LITERATURE SURVEY

2.1 Existing problem

Previous work papers have focused only on a particular imaging modality such as MRI or PET, or on one specific type of dementia only such as AD. Voice recordings of the patients were used as one of the key methods detect Parkinson's disease. Voice recordings cannot be used all time since some patients can't speak

2.2 References

- Minhazul Arefin, Kazi Mojammel Hossen, Rakib Hossen and Mohammed Nasir Uddin,"Parkinson’s Disease Handwriting Detection Using FCNN",2022
- Nihar Ranjan, Divya Umesh Kumar, Vaishnavi Dongare,Kiran Chavan,Yuvraj Kuwar"Diagnosis of Parkinson Disease using Handwriting Analysis",2022
- Isham Vatsaraj, Dr. Gajanan Nagare, “Early Detection of Parkinson’s disease using Contrast Enhancement and CNN”,2021.
- Jinse Park, Sabyasachi Chikraborti, “Parkinson’s Disease Detection From Sprial and Wave Drawings using CNN: A Multistage classifier Approach”, 2020.
- DonatoImpedovo and Giuseppe Pirlo, Dynamic Handwriting Analysis for the Assessment of Neurodegenerative: A Pattern Recognition Perspective VOL. 12, 2019
- Tim Hahm,Max A.Little, Marjan J. Faber , “Freezing of gait and fall detection in Parkinson’s disease using wearable sensors : a systematic review” 2018.

2.3 Problem Statement Definition

Problem Statement

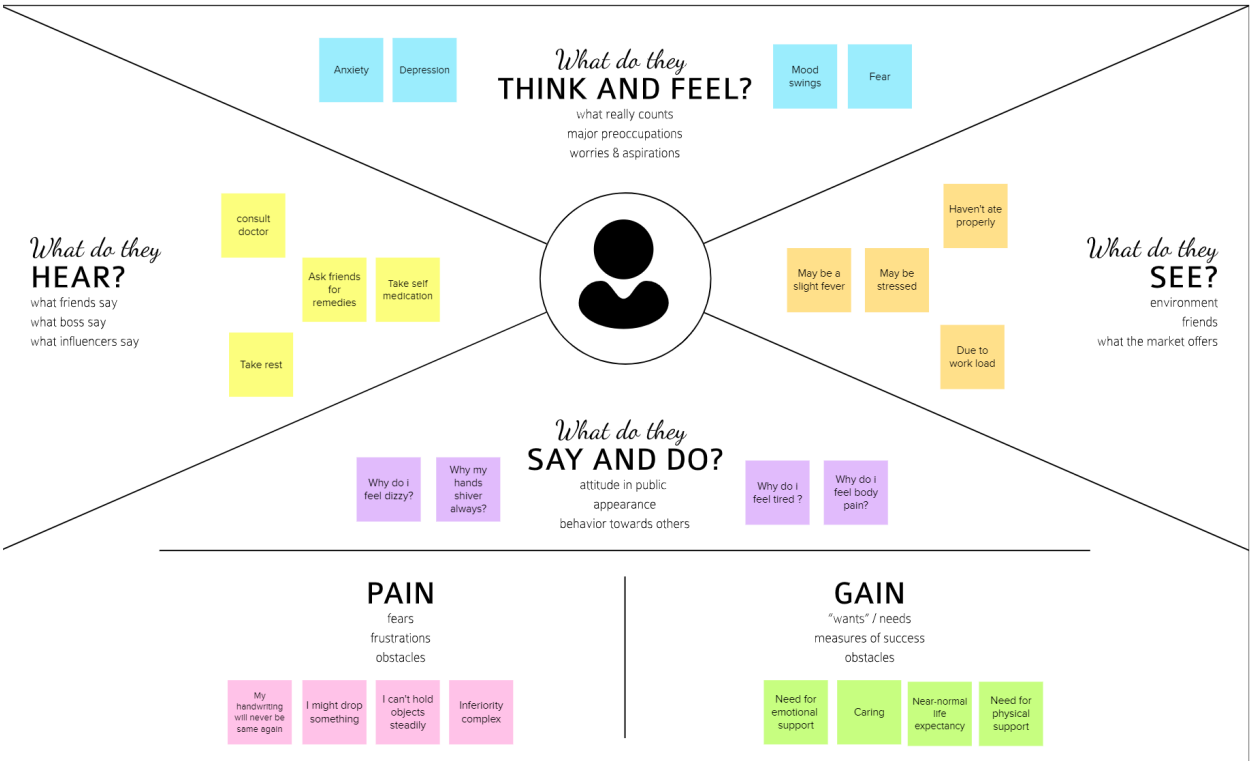
Sunil is a Mathematics teacher in a School who finds difficult to write on board because his hands tremble a lot. Vijay needs to find the disorder in him so that he can take proper medication and write on board like he previously did. Something, which should fit into the palm of his hands which he could carry anywhere, access from anywhere, something digital such as an application that detects the disorder hassle-free which could be accessed from any device such as smart phones and computers.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Sunil	Write on board	my hands tremble	Parkinson's disease	Frustated
PS-2	Rahul	Walk	Loses my Balance	Parkinson's disease	Fearful
PS-3	Anitha	Sing	Experiencing undesirable voice changes	Parkinson's disease	Depressed
PS-4	Johnny	Boxing	Stiffness in muscles	Parkinson's disease	Stressful
PS-5	Sarath	Manager	numbness in one arm	Parkinson's disease	low

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

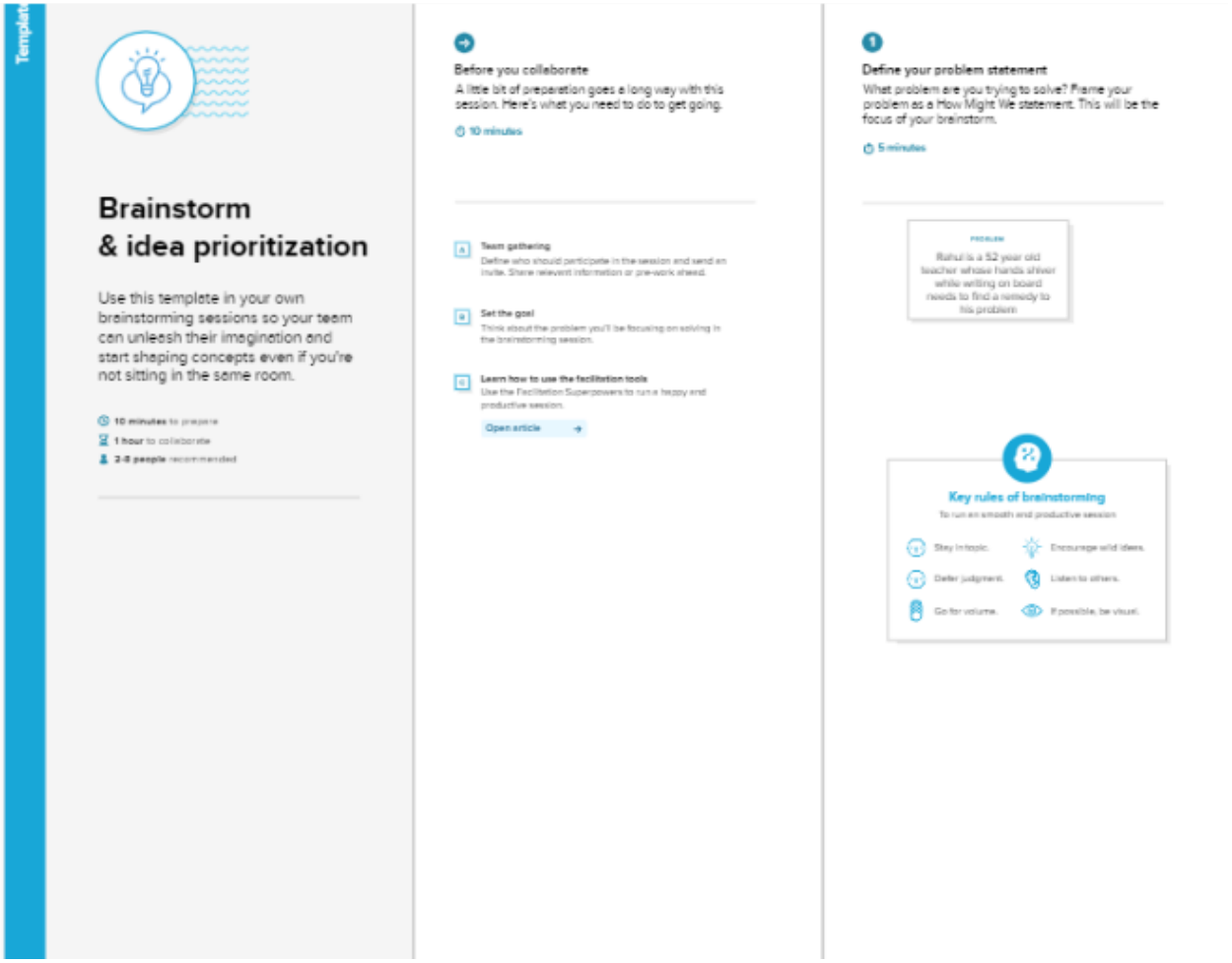
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



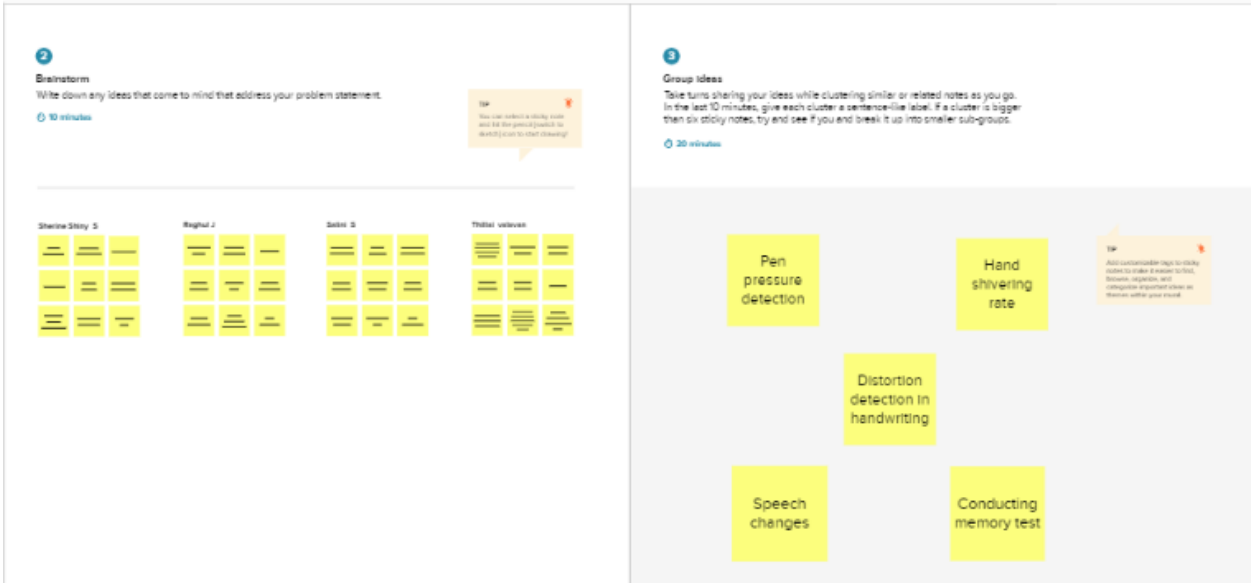
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

3.3 Proposed Solution

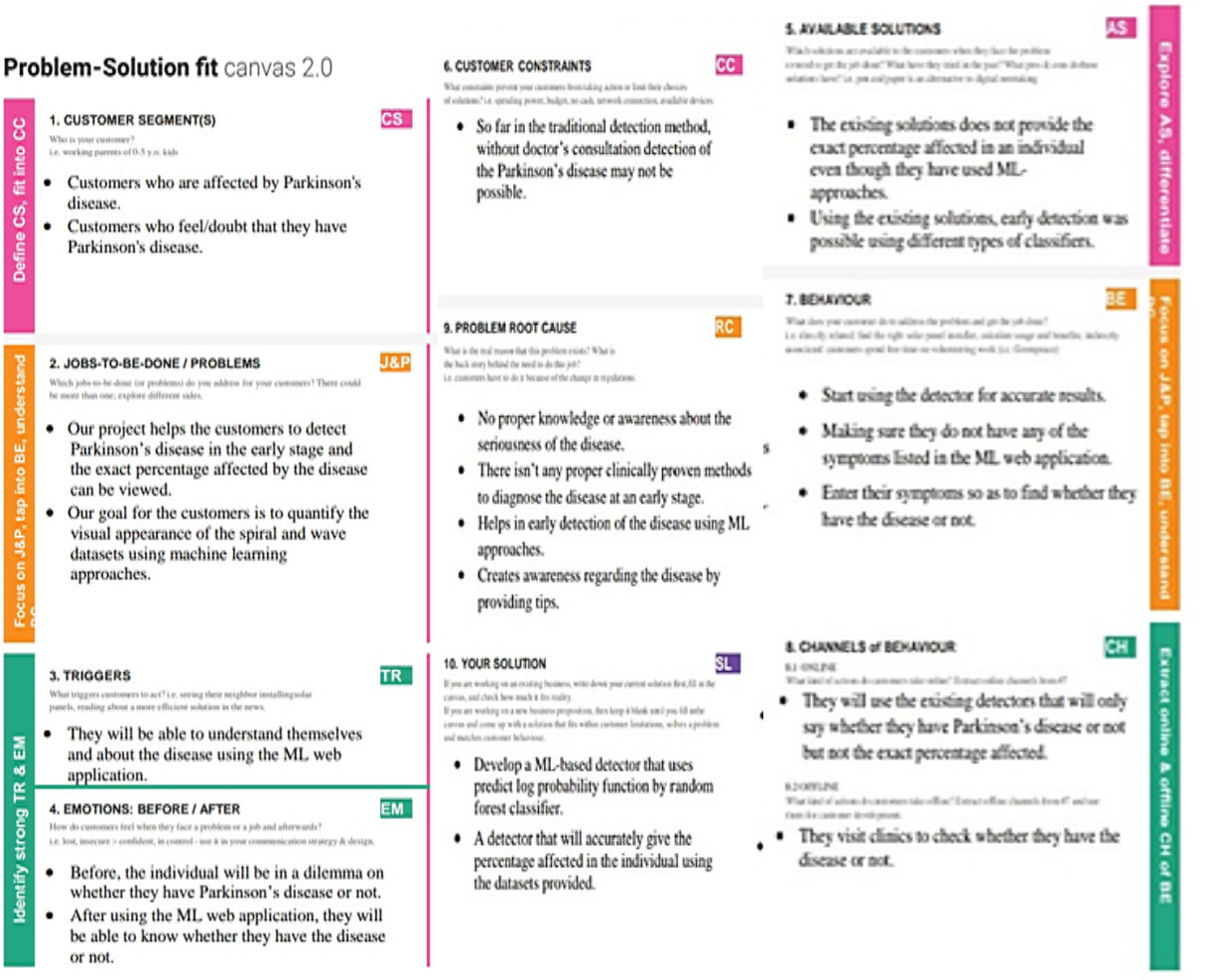
This proposed system would diagnose a Parkinson patient using the drawings of spirals and waves which is taken as input to the system. The dataset consists of drawings of both healthy people and Parkinson patients for analysis. This system uses Histogram of Oriented Gradients (HOG) and Random Forest Classifier for the classification of the input images.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Sunil is a 52 years old teacher who find difficult to write on board as his hands shiver a lot need to find a right treatment for his problem.
2.	Idea/Solution description	This project uses Histogram of Oriented Greadients(HOG) along with Random Forest Classifier to automatically detect Parkinson's disease in hand drawn images of spiral and waves.
3.	Novelty/Uniqueness	To increase the accuracy of corner in images, HOG algorithm is introduced.
4.	Social impact/Customer Satisfaction	To detect the disease and control the effects of the disease in early stage.
5.	Business Model (Revenue Model)	Payers-Next generation managed care models.

6.	Scalability of the Solution	Any additional functionality, such as the employment of a chatbot to assist consumers, may be introduced at anytime and anyplace. It may be accessed simultaneously.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Forms
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Login	Login to landing page using email and password.
FR-4	Image Uploading and Processing	Select image from the system and upload to the application.
FR-5	Identification/Prediction	Prediction of the status of condition based on the hand drawn image uploaded.
FR-6	Accuracy	Provide the accuracy of the performed classification.
FR-7	Medical Suggestions	Suggest necessary remedy, treatment objectives, specialist contacts based on user consent.
FR-8	Obtaining the data	Ability to view the reports as case details.

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

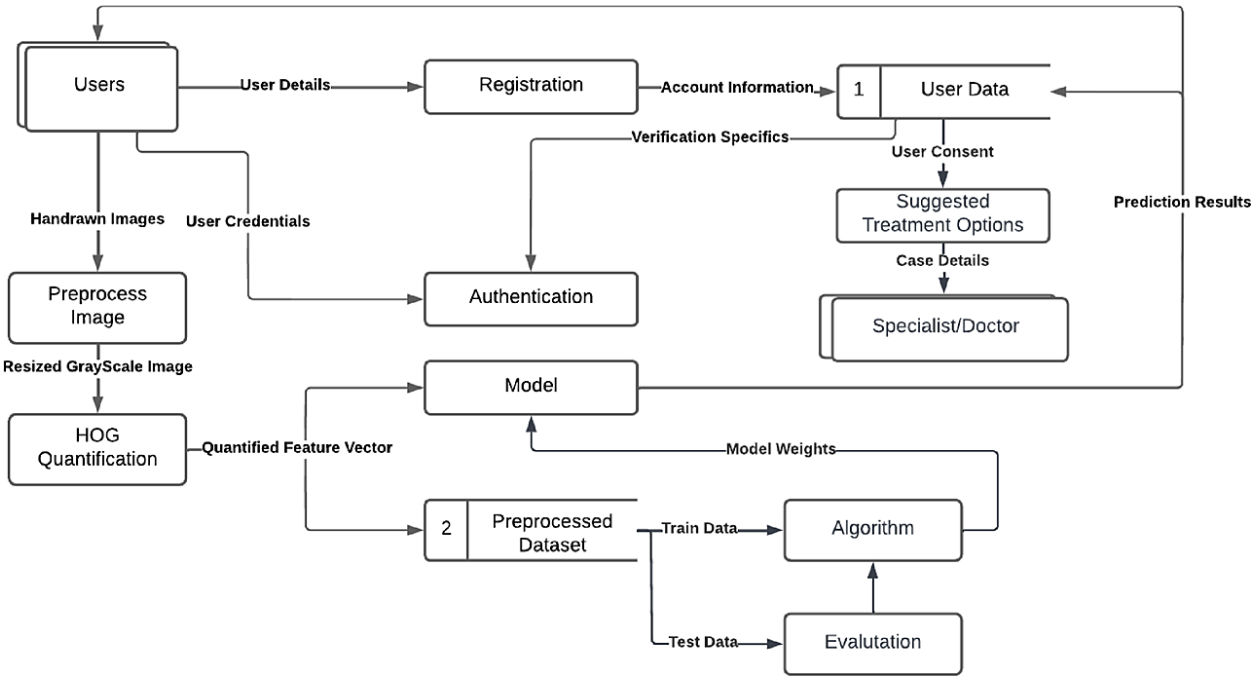
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application is self-intuitive and interactive. Even people with the

		minimal technological awareness can handle it with ease.
NFR-2	**Security**	The user details are bound to the user profile and are secured.
NFR-3	**Reliability**	The application is monitored periodically in terms of its constant prediction ability, quality and availability to the user.
NFR-4	**Performance**	The light weight model used in prediction will enable the accurate prediction of the result.
NFR-5	**Availability**	The application is active throughout the day, with constant maintenance checks every week.
NFR-6	**Scalability**	Application performs well under an increased workload.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

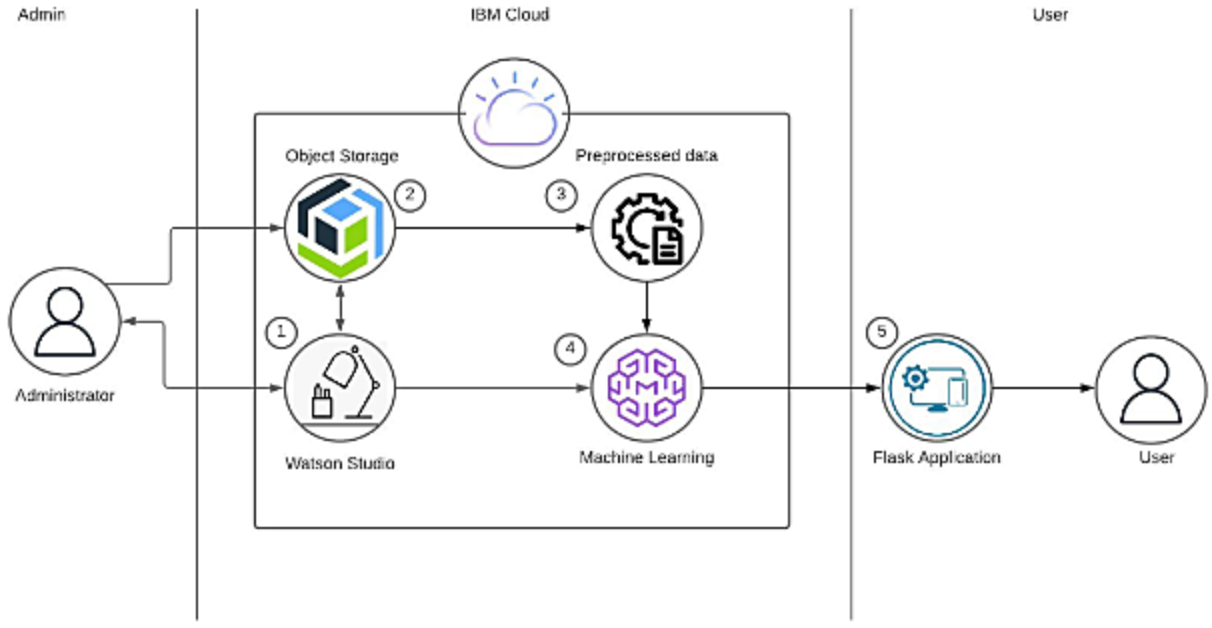
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



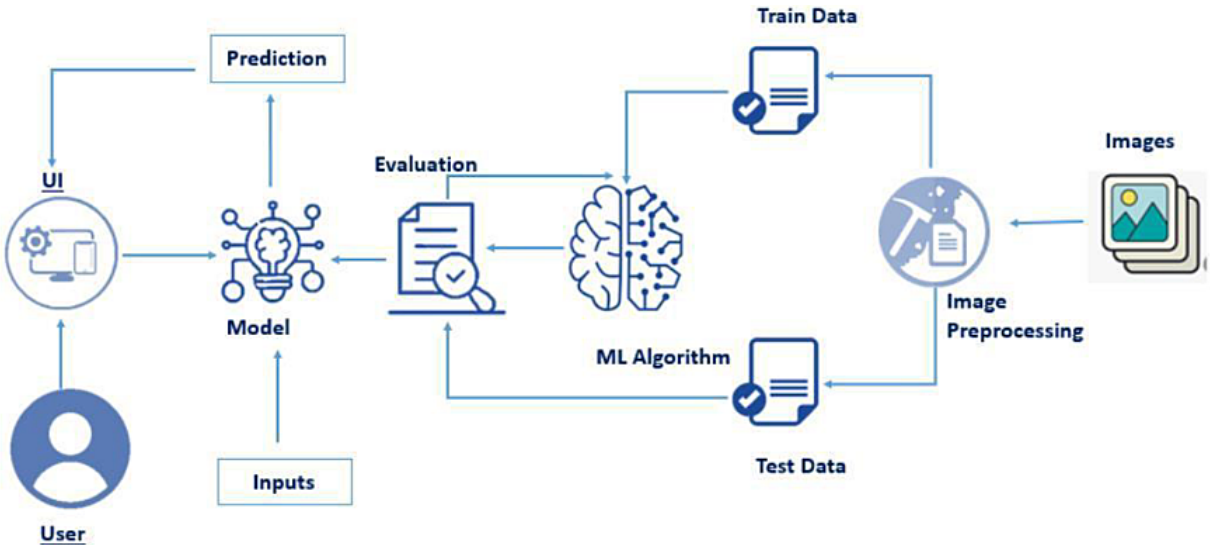
5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Technology Architecture



Solution Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Public user)	Registration	USN-1	As a user, I can register for the application by entering my username, email,phone number, address, occupation, 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	I can receive confirmation email	Medium	Sprint - 1

			for the app			
	Login	USN-3	As a user, I can log into the application by entering email & password	I can log into my account and check my details	High	Sprint - 1
	Image Uploading and Processing	USN-4	As a user, I can upload the image to the application for the purpose of diagnosis	I can successfully upload the image from system images gallery.	High	Sprint - 2
	Identification / Prediction	USN-5	As a user, I can verify with the application that the image is used for the prediction.	I can view the results of the prediction	High	Sprint - 2
	Accuracy	USN-6	As a user, I can understand the accuracy of the prediction that the model has produced	I can see the accuracy with which the model has predicted.	Medium	Sprint - 3
	Medical Suggestions	USN-7	As a user, I would like to take further steps in treatment of the condition.	I can see specialist clinics and medicines suggestions	High	Sprint - 3
Customer (Medical Expert)	Identification / Prediction	USN-8	As a user, I can use the application for preliminary analysis	I can view the results of the prediction with case details	High	Sprint - 2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers,research publications etc.	3 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	10 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	10 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	10 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	7 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	7 OCTOBER 2022
Customer Journey	Prepare the customer jour maps to understand the interactions & experiences with the application.	7 OCTOBER 2022

Data Flow Diagrams	Draw the data flow diagrams and submit for review.	15 OCTOBER 2022
Technology Architecture	architecture diagram.	15 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing	IN PROGRESS..

6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation

SPRINT	FUNTIONAL REQUIREMENT (EPIC)	USER STORY NUMBER	USER STORY/ TASK	STORY POINTS	PRIORITY
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High
Sprint 1		USN-2	As a user, I will receive confirmation email once I	2	Medium

			have registered for the application		
Sprint 2		USN-3	As a user, I can register for the application through mobile number	3	High
Sprint 2		USN-4	As a user. I will receive confirmation SMS	3	High
Sprint 2	Login	USN-5	As a user, I can log into the application by entering login credentials	3	High
Sprint 3	Dashboard	USN-6	As a user, I can upload my images and get my details of skin diseases	3	High
Sprint 1	Logout	USN-7	As a user, I can logout successfully	2	Medium
Sprint 4	Feedback	USN-8	A customer care executive, I can able to interact	2	Medium

			with all the customer and get their feedback which is used to enhance the scope of the project.		
Sprint 3	Image processing localization	USN-9	The uploaded image is preprocessed and fed into the trained HOG model	3	High
Sprint 4	Classification and prediction	USN-9	HOG model classify and predict the type of disease	3	High
Sprint 4	Report generation	USN-10	Based on the prediction of Parkinson Disease, the health care report generated to provide feedback.	2	Medium

Project Tracker, Velocity & Burndown Chart:

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)

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

Average Velocity = Story Points per Day

Sprint Duration = Number of (Duration) days per Sprint

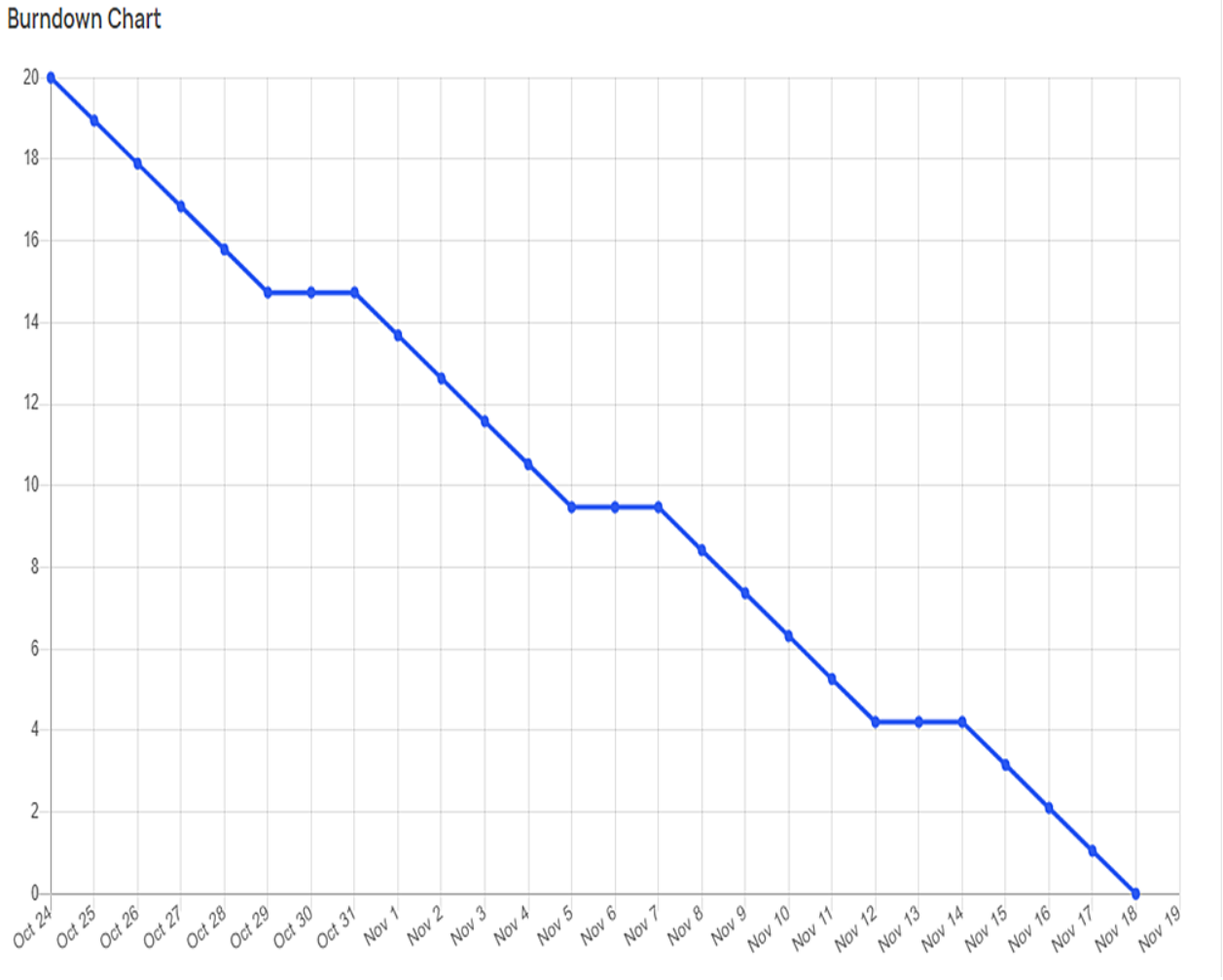
Velocity = Points per Sprint

AV = **20** / ≈ 4

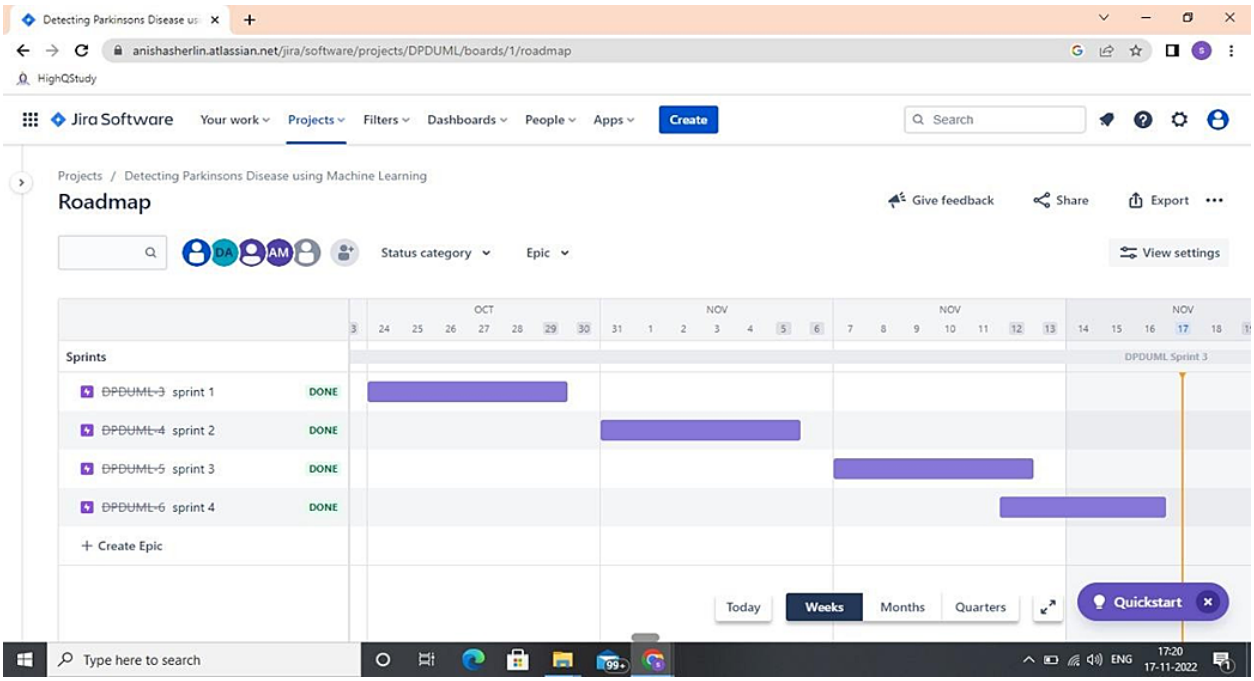
Therefore, the AVERAGE VELOCITY IS 4 POINTS PER SPRINT

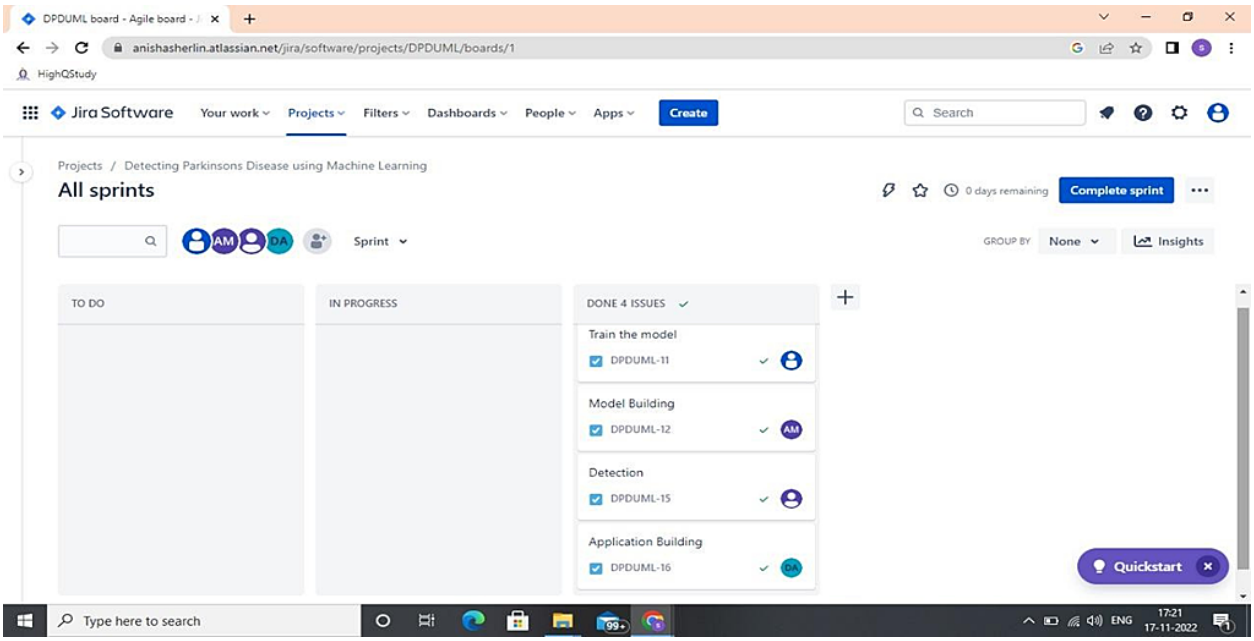
Burndown Chart:

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





7. CODING & SOLUTIONING

7.1 Feature 1

Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is a feature descriptor used in image processing, mainly for object detection.The principle behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions.

Code:

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

7.2 Feature 2

Random Forest Classifier

Random forest is a supervised learning algorithm. Random forest consists of large individual decision trees that operate as an ensemble.The decision tree is a ordered structure that is built using the features of a data set. Every node of the decision tree is break according to a measure associated with a subset of the features. The random forest is a collection of decision trees that are linked with a set of bootstrap samples that are generated from the original data set. The nodes are subdivided based on the entropy of a selected subset of the features.

Code:

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
```

7.3 Database Schema

Database: user_login

Table: accounts

ColumnName	DataType	Allow Nulls
id	int(11)	Not Null
username	varchar(50)	Not Null
password	varchar(255)	Not Null
email	varchar(100)	Not Null

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N	BUG ID	Executed By
LoginPage_TC_003	Functional	Home page	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	Username: sherine shiny password: shiny2001	User should navigate to user account homepage	Working as expected	Pass	Takes more time to load	Y	3	Salini S, Raghul J
LoginPage_TC_004	Functional	Login page	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: sbi password: sbi2001	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	No feature for changing password	Y	4	Raghul J, Thillaivalavan A S
LoginPage_TC_004	Functional	Login page	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	Username: rahul password: rahul	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	No feature for changing password	Y	5	Salini S, Raghul J
LoginPage_TC_005	Functional	Login page	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: jenny password:jenny	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	No feature for changing password	Y	6	Sherine Shiny S, Salini S
PredictionPage_TC_001	Functional	Prediction Pag	Verify user can upload image fil	1.Enter the valid URL(http://127.0.0.1:5000) 2..Enter Valid username and password in text box 3.Click on login button 4.Click Choose Button 5.Upload the image file	image file	display of the uploaded image	Working as expected	Pass	No drag and drop feature of image file	Y	7	hillaivalavan A S, Salini S
PredictionPage_TC_002	Functional	Prediction Pag		1.Enter the valid URL(http://127.0.0.1:5000) 2..Enter Valid username and password in text box 3.Click on login button	other format file	please upload an image file	Working as expected	Pass	nil	Y	nil	aghul J, Thillaivalavan A

8.2 User Acceptance Testing

Defect Analysis:

This report shows the numberof resolved or closed bugs at each severity level, and howtheywere 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

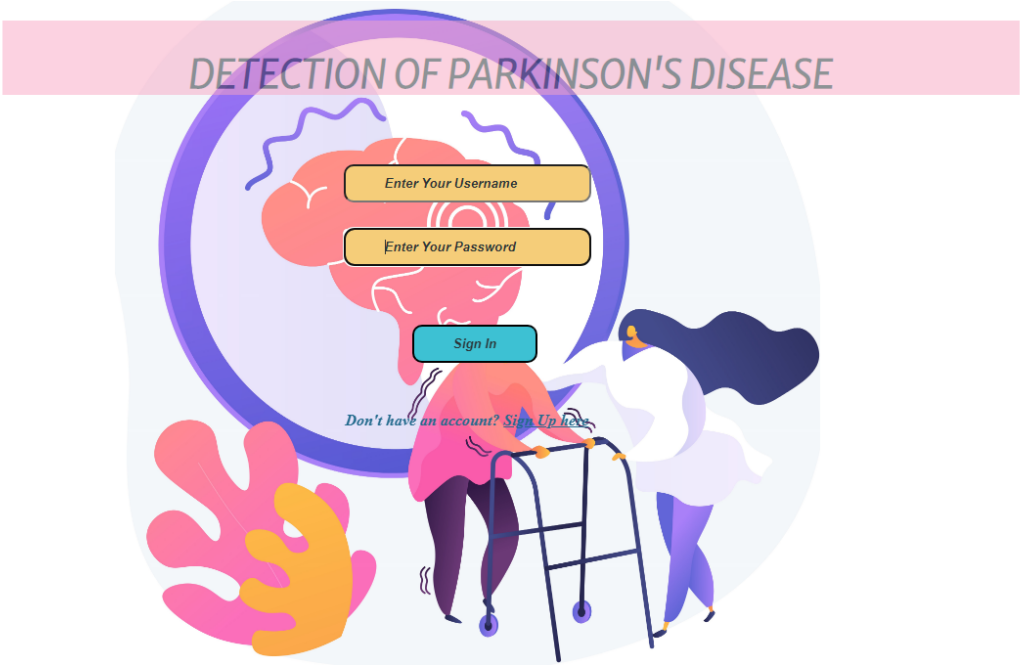
This report shows the number of test cases that have passed, failed and untested

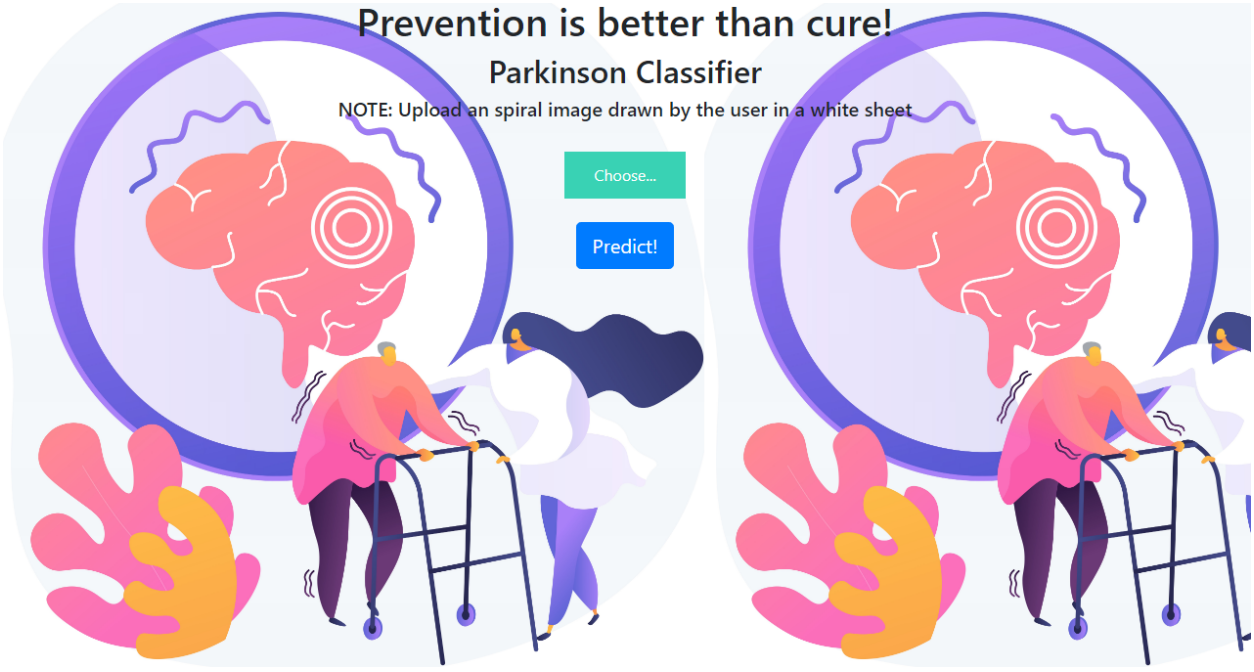
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	53	0	0	53
Security	2	0	0	2
Outsource Shipping	4	0	0	4
Exception Reporting	8	0	0	8

Final ReportOutput	5	0	0	5
Version Control	2	0	0	2

9. RESULTS

In this project,we found that Parkinsons diseasecan be detected using the value's obtained from uploaded image of spiral.

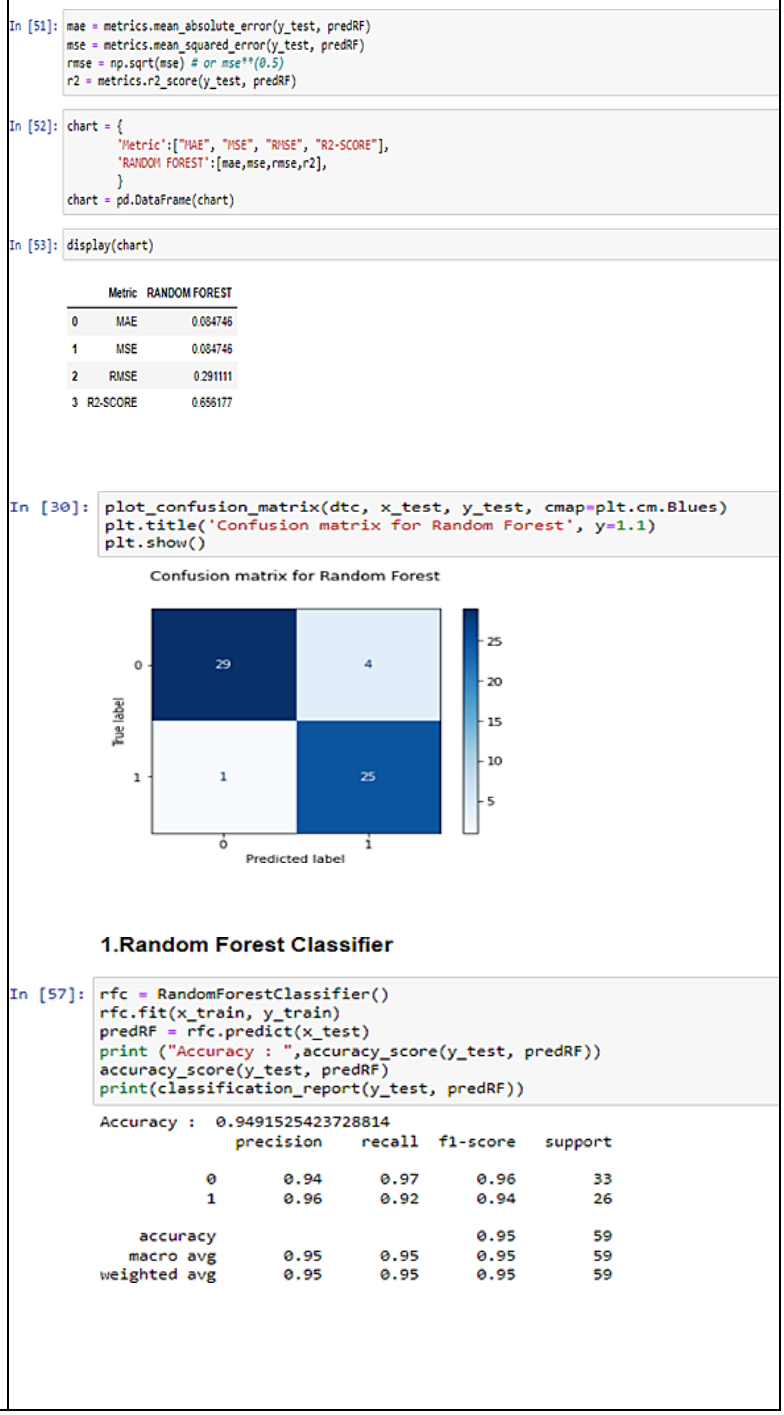



Prevention is better than cure!
Parkinson Classifier
NOTE: Upload an spiral image drawn by the user in a white sheet
Choose...
Predict!


Prevention is better than cure!
Parkinson Classifier
NOTE: Upload an spiral image drawn by the user in a white sheet
Choose...
Prediction : healthy


Prevention is better than cure!
Parkinson Classifier
NOTE: Upload an spiral image drawn by the user in a white sheet
Choose...
Prediction : parkinson

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()
```

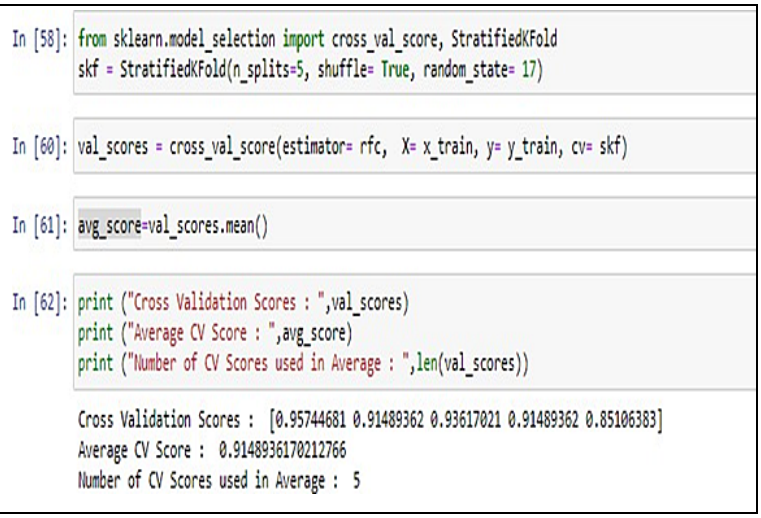

1.Random Forest Classifier

```
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]	

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

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

DISADVANTAGES

- Handicapped persons cannot use this application.

11. CONCLUSION

Previous work papers have focused only on a particular imaging modality such as MRI or PET, or on one specific type of dementia only such as AD. The proposed method aimed to cover a broader space of imaging and machine learning technologies for mental illness diagnostics such that researchers in the field could readily identify the state of the art in the domain. Moreover, in this we emphasize the importance of early detection and prediction of Parkinson's disease, such that treatment and support can be provided to patients as soon as possible and the effects of the disease can be decreased.

12. FUTURE SCOPE

- We can add drag and drop feature for uploading the image file
- We can add a chatbot to assist the users
- We can add feature for live consultation of doctors to help the users
- We can add digital board for drawing instead of uploading file.

13. APPENDIX

Source Code

Parkinson_disease_Predicition.ipynb

Image Pre-Processing

Import the necessary Libraries

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from skimage import feature
from imutils import build_montages
from imutils import paths
import numpy as np
import cv2
import os
import pickle
```

Path for train and test data

```
trainingpath=r"dataset/spiral/training"
testingpath=r"dataset/spiral/testing"
```

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
```

Loading Train Data and Test Data

```
def load_split(path):
    imagePaths = list(paths.list_images(path))
    data = []
    labels = []

    for imagePath in imagePaths:
        label = imagePath.split(os.path.sep)[-2]
```

```
        image = cv2.imread(imagePath)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))

        image=cv2.threshold(image,0,255,cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

        features = quantify_image(image)

        data.append(features)
        labels.append(label)

    return (np.array(data), np.array(labels))
```

Load the train and test data

```
print("[INFO] loading data...")
(X_train, y_train) = load_split(trainingpath)
(X_test, y_test) = load_split(testingpath)
```

Label Encoding

```
le = LabelEncoder()
y_train = le.fit_transform(y_train)
y_test = le.transform(y_test)
print(X_train.shape,y_train.shape)
```

Model Building

Training The Model

```
print("[INFO] training model")
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
```

Testing The Model

```
testingpath=list(paths.list_images(testingpath))
idxs=np.arange(0,len(testingpath))
idxs=np.random.choice(idxs,size=(25,),replace=False)
images=[]
for i in idxs:
    image=cv2.imread(testingpath[i])
    output=image.copy()
```

```
    # load the input image,convert to grayscale and resize

    output=cv2.resize(output,(128,128))
    image=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    image=cv2.resize(image,(200,200))
    image=cv2.threshold(image,0,255,cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

    #quantify the image and make predictions based on the  extracted feature using last trained
random forest
    features=quantify_image(image)
    preds=model.predict([features])
    label=le.inverse_transform(preds)[0]
    #the set of output images
    if label=="healthy":
        color=(0,255,0)
    else:
        color=(0,0,255)

    cv2.putText(output,label,(3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)
    images.append(output)

#creating a montage
montage=build_montages(images,(128,128),(5,5))[0]
cv2.imshow("Output",montage)
cv2.waitKey(0)
```

Model Evaluation

```
predictions = model.predict(X_test)

cm = confusion_matrix(y_test, predictions).flatten()
print(cm)
(tn, fp, fn, tp) = cm
accuracy = (tp + tn) / float(cm.sum())
print(accuracy)
```

Save The Model

```
pickle.dump(model,open('parkinson.pkl','wb'))
```

local_deployment.ipynb

Import the necessary libraries

```
from flask import Flask, render_template, request, redirect, url_for,
session,abort
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re
import imghdr
import os
from werkzeug.utils import secure_filename
from flask import send_from_directory
import pickle
import numpy as np
from PIL import Image
import cv2
from skimage import feature
import os.path

app = Flask(__name__)
app.config['MAX_CONTENT_LENGTH'] =1024* 1024 * 1024
app.config['UPLOAD_EXTENSIONS'] = ['.jpg', '.png', '.gif']
app.config['UPLOAD_PATH'] = 'uploads'
def validate_image(stream):
    header = stream.read(1024)
    stream.seek(0,2)
    format = imghdr.what(None, header)
    if not format:
        return None
    return '.' + (format if format != 'jpeg' else 'jpg')


app.secret_key = 'your secret key'

app.config['MYSQL_HOST'] = 'localhost'
```

```
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'user_login'

mysql = MySQL(app)

@app.route('/')
@app.route('/front', methods =['GET', 'POST'])
def login():
        msg = ''
        if request.method == 'POST' and 'username' in request.form and
'password' in request.form:
                username = request.form['username']
                password = request.form['password']
                cursor =
mysql.connection.cursor(MySQLdb.cursors.DictCursor)
                cursor.execute('SELECT * FROM accounts WHERE username =
% s AND lpassword = % s', (username, password, ))
                account = cursor.fetchone()
                if account:
                        session['loggedin'] = True
                        session['id'] = account['id']
                        session['username'] = account['username']
                        msg = 'Logged in successfully !'
                        return render_template('predict.html', msg = msg)
                else:
                        msg = 'Incorrect username / password !'
        return render_template('front.html', msg = msg)

@app.route('/logout')
def logout():
        session.pop('loggedin', None)
```

```
        session.pop('id', None)
        session.pop('username', None)
        return redirect(url_for('login'))

@app.route('/register', methods =['GET', 'POST'])
def register():
        msg = ''
        if request.method == 'POST' and 'username' in request.form and
'password' in request.form and 'email' in request.form :
                username = request.form['username']
                password = request.form['password']
                email = request.form['email']
                cursor =
mysql.connection.cursor(MySQLdb.cursors.DictCursor)
                cursor.execute('SELECT * FROM accounts WHERE username =
% s', (username, ))
                account = cursor.fetchone()
                if account:
                        msg = 'Account already exists !'
                elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
                        msg = 'Invalid email address !'
                elif not re.match(r'[A-Za-z0-9]+', username):
                        msg = 'Username must contain only characters and
numbers !'
                elif not username or not password or not email:
                        msg = 'Please fill out the form !'
                else:
                        cursor.execute('INSERT INTO accounts VALUES (NULL, %
s, % s, % s)', (username, password, email, ))
                        mysql.connection.commit()
                        msg = 'You have successfully registered !'
        elif request.method == 'POST':
```

```
            msg = 'Please fill out the form !'
        return render_template('register.html', msg = msg)
@app.route("/upload")
def test():
   return render_template("predict.html")
@app.route('/predict', methods=['GET', 'POST'])
def upload():
   if request.method == 'POST':
      f=request.files['file'] #requesting the file
      basepath=os.path.dirname(os.path.realpath('__file__'))#storing the file
directory
      filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in
uploads folder
      f.save(filepath)#saving the file

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

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

      # Load the input image, convert it to grayscale, and resize
      output = cv2.resize(output, (128, 128))
      image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
      image = cv2.resize(image, (200, 200))
      image = cv2.threshold(image, 0, 255,
        cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

        # Quantify the image and make predictions based on the extracted
features using the last trained Random Forest
```

```
        features = feature.hog(image, orientations=9,
                pixels_per_cell=(10, 10), cells_per_block=(2, 2),
                transform_sqrt=True, block_norm="L1")
        preds = model.predict([features])
        print(preds)
        ls=["healthy","parkinson"]
        result = ls[preds[0]]
        return result
    return None
```

front.html

```
<html>
        <head>
                <meta charset="UTF-8">
                <title>Login</title>
                <link rel="stylesheet" href="{{ url_for('static',
filename='css/cstyle.css') }}">
        </head>
        <body></br></br></br></br></br>
                <div align="center">
                <div align="center" class="border">
                        <div class="header">
                                <h1 class="word">DETECTION OF PARKINSON'S
DISEASE</h1>
                        </div></br></br></br>
                        <h2 class="word">
                                <form action="{{ url_for('login') }}" method="post">
                                <div class="msg">{{ msg }}</div>
                                        <input id="username" name="username"
type="text" placeholder="Enter Your Username" class="textbox"
autocomplete="off"/></br></br>
                                        <input id="password" name="password"
type="password" placeholder="Enter Your Password" class="textbox"
```

```
autocomplete="off"/></br></br></br>
                                <input type="submit" class="btn" value="Sign
In"></br></br>
                        </form>
                    </h2>
                    <p class="bottom">Don't have an account? <a
class="bottom" href="{{url_for('register')}}"> Sign Up here</a></p>
                </div>
                </div>
        </body>
</html>
```

register.html

```
<html>
        <head>
                <meta charset="UTF-8">
                <title> Register </title>
                <link rel="stylesheet" href="{{ url_for('static',
filename='css/cstyle.css') }}">
        </head>
        <body></br></br></br></br></br>
                <div align="center">
                <div align="center" class="border">
                        <div class="header">
                                <h1 class="word">Register</h1>
                        </div></br></br></br>
                        <h2 class="word">
                                <form action="{{ url_for('register') }}"
method="post">
                                <div class="msg">{{ msg }}</div>
                                        <input id="username" name="username"
type="text" placeholder="Enter Your Username" class="textbox"
autocomplete="off"/></br></br>
```

```
                                    <input id="password" name="password"
type="password" placeholder="Enter Your Password" class="textbox"
autocomplete="off"/></br></br>
                                    <input id="email" name="email" type="text"
placeholder="Enter Your Email ID" class="textbox"
autocomplete="off"/></br></br>
                                    <input type="submit" class="btn" value="Sign
Up"></br>
                                </form>
                            </h2>
                            <p class="bottom">Already have an account? <a
class="bottom" href="{{url_for('login')}}"> Sign In here</a></p>
                    </div>
                    </div>
            </body>
</html>
```

predict.html

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Predict</title>
    <link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></scri
pt>
    <script
src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></scrip
```

```
t>
   <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
   <style>
      body {
        background-image: url('http://formsaws.s3-website.ap-south-
1.amazonaws.com');
        background-size: contain;
        max-width: 100%;
        max-height: 100%;
        overflow: hidden;
      }
      </style>
</head>
<body>
   <h1><center>Prevention is better than cure!</center></h1>
      <h2><center>Parkinson Classifier</center></h2>
      <h5><center>NOTE: Upload an spiral image drawn by the user in a
white sheet</center></h5>
      <div class="container">
        <center> <div id="content" style="margin-top:2em">{% block content
%}{% endblock %}</div></center>
      </div>
      <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>

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

    </div>
    <footer>
        <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
    </footer>
</body>
```

cstyle.css

```
@import
url('https://fonts.googleapis.com/css2?family=Merriweather+Sans:ital,wgh
t@1,300&family=Pacifico&display=swap');
@import
```

```
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400
;500;600;700&display=swap');
.header{
    padding: 3px 120px;
    width: 800px;
    height: 70px;
    background-color: #f8a5c2;
    opacity: 0.5;
    margin-top: -5rem;
}
.head{
    padding: 3px 120px;
    width: 800px;
    height: 70px;
    background-color: #f8a5c2;
    opacity: 0.5;
    margin-top: -2rem;
}
.msg{
    margin-top: -1rem;
}

.border{
    padding: 80px 50px;
    width: 950px;
    height: 475px;
    border-radius: 0px;
}


body {
    background-image: url("http://formsaws.s3-website.ap-south-
```

```
1.amazonaws.com");
    background-size: contain;
    background-position: center;
    max-width: 170%;
    max-height: 100%;
    background-repeat: no-repeat;
    overflow: hidden;
    display: flex;
    align-items: center;
    justify-content: center;
}

.btn {
    padding: 10px 40px;
    background-color: #3dc1d3;
    color: #2d3436;
    font-style: oblique;
    font-weight: bold;
    border-radius: 10px;
    margin-top: -1rem;
    position: relative;
    margin-left: 1rem;
}
.but {
    padding: 15px 32px;
    background-color: #3dc1d3;
    color: #2d3436;
    font-style: oblique;
    font-weight: bold;
    border-radius: 10px;
    margin-top:35rem;
    margin-right: 10rem;
```

```
    position: absolute;
}

.textbox{
    padding: 10px 40px;
    background-color: #f5cd79;
    color: #2d3436;
    border-radius: 10px;
    margin-bottom: -1.5rem;

}

::placeholder {
    color: #2d3436;
    opacity: 1;
    font-style: oblique;
    font-weight: bold;
}

.word{
    color: #1e272e;
    font-family: 'Merriweather Sans', sans-serif;
    font-weight: bold;
    font-size: 40px;
}

.bottom{
    color: #236B8E;
    font-style: oblique;
    font-weight: bold;
    margin-top: -2.5rem;
}
```

```
.drag-area{
   border: 2px dashed #2d3436;
   height: 300px;
   width: 600px;
   border-radius: 5px;
   display: flex;
   align-items: center;
   justify-content: center;
   flex-direction: column;
   margin-top: 3rem;
 }
 .drag-area.active{
   border: 1px solid #2d3436;
 }
 .drag-area .icon{
   font-size: 100px;
   color: #fff;
 }
 .drag-area header{
   font-size: 30px;
   font-weight: 500;
   color: #2d3436;
   font-family:"Poppins", sans-serif;
 }
 .drag-area span{
   font-size: 25px;
   font-weight: 500;
   color: #2d3436;
   margin: 10px 0 15px 0;
   font-family:"Poppins", sans-serif;
 }
 .drag-area button{
```

```
  padding: 10px 25px;
  font-size: 20px;
  font-weight: 500;
  border: none;
  outline: none;
  background: #3dc1d3;
  color: #5256ad;
  border-radius: 5px;
  cursor: pointer;
}
.drag-area button{
  padding: 10px 25px;
  font-size: 20px;
  font-weight: 500;
  border: none;
  outline: none;
  background: #3dc1d3;
  color: #2d3436;
  border-radius: 5px;
  cursor: pointer;
}
.drag-area img{
  height: 100%;
  width: 100%;
  object-fit: contain;
  border-radius: 5px;
}

.button {
  border: none;
  color: #2d3436;
  background: #3dc1d3;
```

```
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    cursor: pointer;
    position: relative;
    bottom: 10px;
    border-radius: 5px;
    font-family: "Poppins", sans-serif;
    font-size: 20px;
    margin-top: 10rem;
    margin-right: -8rem;
  }
```

main.css

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

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}
```

```
input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

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

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

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

main.js

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

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

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

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

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

});
```

GitHub & Project Demo Link

GitHub: https://github.com/IBM-EPBL/IBM-Project-2835-1658483958

Project Demo Link:https://drive.google.com/file/d/1wa3dslzwZgNx05486NawZd3k1aw62RfU/view