Fertilizers Recommendation System for Disease Prediction

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID26231

NIRANJANA SR - 211519205104
KIRTHANA P - 211519205079
HARSHA VARDINI G - 211519205060
HARINI M - 211519205057
NOWZIYA BANU Y - 211519205108

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

PANIMALAR INSTITUTE OF TECHNOLOGY,

CHENNAI.

1.INTRODUCTION

1.1 Overview In this project, two datasets named fruits and vegetables are collected. The collected datasets are trained and tested using a deep learning neural network called Convolutional Neural Networks (CNN). First, the fruit set is trained and then tested using CNN. It has 6 classes and all classes are trained and tested. Second, a vegetable dataset is trained and tested. The software used for training and testing datasets is Python. All Python code is first written in the Jupyter notebook that comes with Anaconda Python, and then the code is tested in the IBM cloud. Finally, a web framework is designed with the help of Flask and the Python library. 2 html files are created in the templates folder along with their associated files in the static folder. The python program 'app.py' used to link to these two web pages is written in Spyder-Anaconda python and tested.

1.2 Purpose This project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

2.1.1 Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at their earliest. Plant pathologists can analyze digital images using digital image processing for the diagnosis of plant diseases. The proposed method uses SVM to classify tree leaves, identify the disease and suggest fertilizer. The proposed method is compared with the existing CNN - based leaf disease prediction. The proposed SVM technique gives a better result when compared to the existing CNN.

Advantages: This project uses a Support Vector Machine to identify diseases in leaves. The accuracy of this system is more than compared to CNN. The F- measure for the given set of images is high. The proportion of the actual positives is identified correctly.

Disadvantages: It doesn’t provide 100% accuracy as the system is implemented with existing datasets. This system cannot identify diseases in other plant organs. It can cause memory leaks over time as it is developed using dot NET.

2.1.2 The project proposes a deep learning-based model that will be trained with photos of healthy and diseased crop leaves from a dataset. The model will achieve its goal by categorizing photos of leaves into unhealthy categories based on defect patterns. The generated datasets of diseased and healthy leaves are combined and trained under Random Forest to identify the sick and healthy pictures.

Advantages: Deep Learning methods were used in this project to correctly detect plant diseases. This project mainly detects and distinguishes between a healthy plant and different diseases. It provides suitable remedies to cure the disease.

Disadvantages: The system is not robust as it contains only one image dataset. There is no availability of an application or website for the farmers to work. This project does not apply to many crops so the users may not get a suitable remedy.

2.1.3 Displayed by Rakesh Kumar, M.P. Singh, Prabhat Kumar, and J.P. Singh proposed the utilization of seven machine learning methods, for example, ANN, SVM, KNN, Decision Tree, Random Forest, GBDT, and Regularized Gradient Forest for crop choice. The framework is intended to recover every one of the yields sowed and time of development at a specific time. The yield rate of each harvest is acquired and the crops giving higher yields are chosen. The framework additionally proposes an arrangement of crops to be planted to get higher yields.

Advantages: The prediction and diagnosing of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

Disadvantages: Some of the issues in these approaches include the impact of background data on the final picture, optimization of the methodology for a specific plant leaf disease, and automation of the technique for continuous automated monitoring of plant leaf diseases in real-world field circumstances.

2.1.4 In this paper, we propose a user-friendly web application system based on machine learning and web scraping. With this system, we are successfully able to provide several features - crop recommendation using the Random Forest algorithm, fertilizer recommendation using a rule-based classification system, and crop disease detection using the model on leaf images.

Advantages: For crop recommendation and fertilizer recommendation, we can provide the availability of the same on popular shopping websites, and possibly allow users to buy the crops and fertilizers directly from our application.

Disadvantages: To provide fine-grained segmentations of the diseased portion of the dataset. this is not possible due to the lack of such data. However, in our application, we can integrate a segmentation annotation tool where the users might be able to help us with the lack. Also, we can use some unsupervised algorithms to pinpoint the diseased areas in the image. We intend to add these features and fix these gaps in our upcoming work.

2.2 REFERENCES:

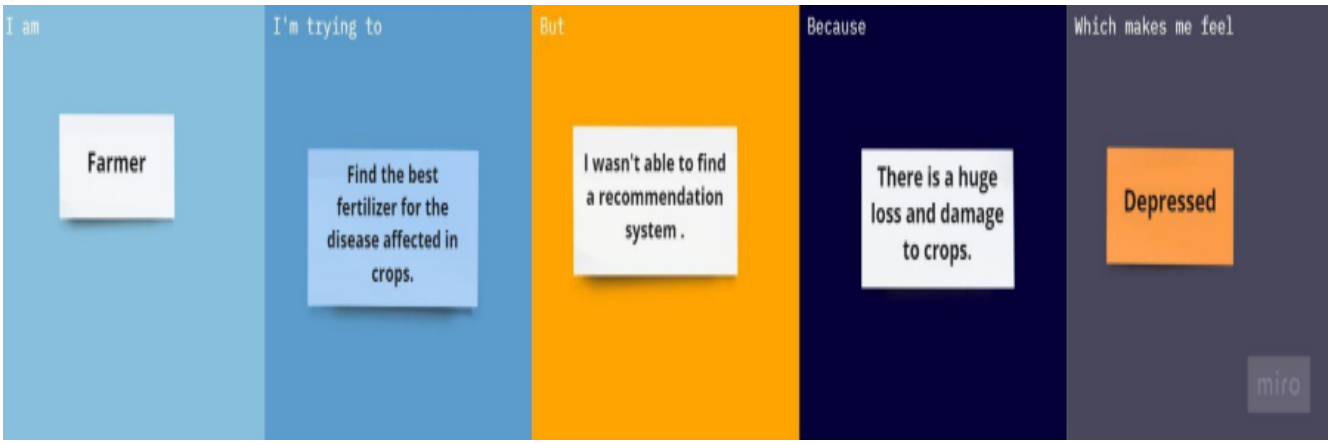
[1]. R Indumathi Leaf Disease Detection and Fertilizer Suggestion", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 29-30 March 2019, DOI: 10.1109/ICSCAN.2019.8878781.

[2]. P. Pandi Selvi, P. Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System", International Journal of Engineering Trends and Applications (IJETA) – Volume 8 Issue 2, Mar-Apr 2021.

[3]. H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3, "IoT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395-005.

2.3 PROBLEM STATEMENT DEFINITION:

Agriculture is the backbone of the Indian economy.Managing agricultural nutrients to provide a safe food supply and secure the environment remains one of the immense challenges of the 21st century.Balanced fertilizer application is not only essential for producing top quality crops in high yields but also for environmental sustainability**.**



3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION AND BRAINSTORMING:

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

Template

Fertilizers Recommendation System for Disease Prediction

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

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

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

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

5 minutes

PROBLEM

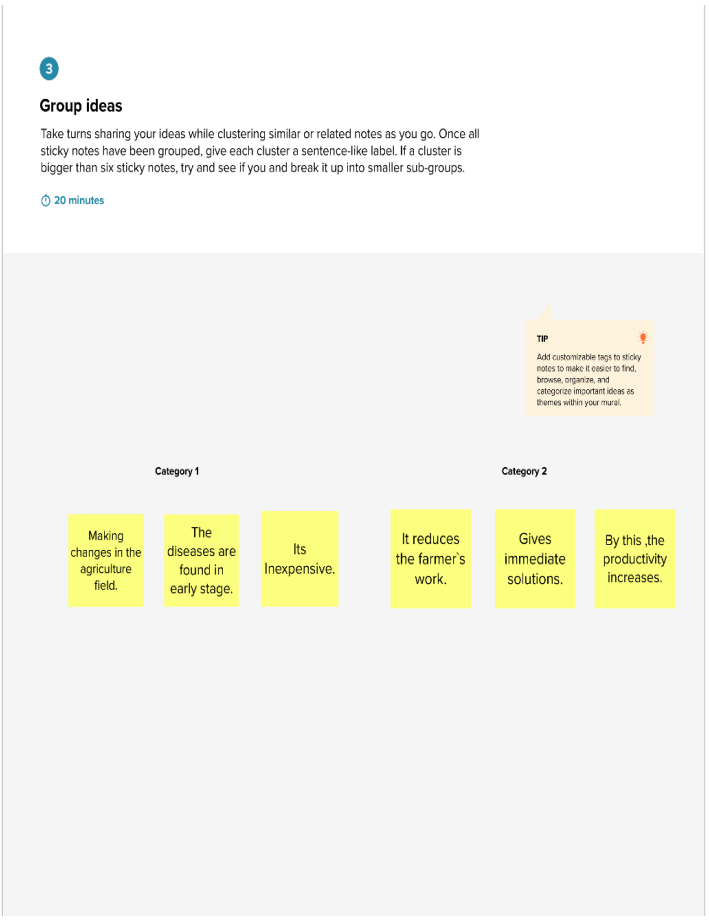
How might we [your problem statement]?

Key rules of brainstorming

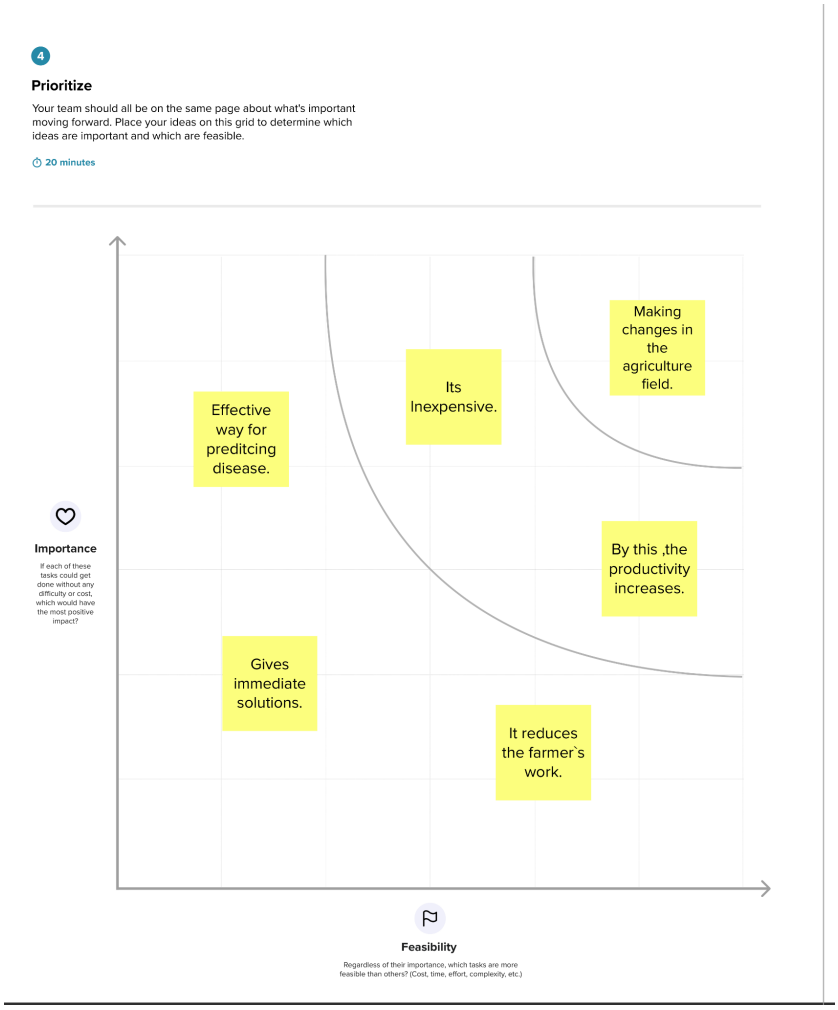
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

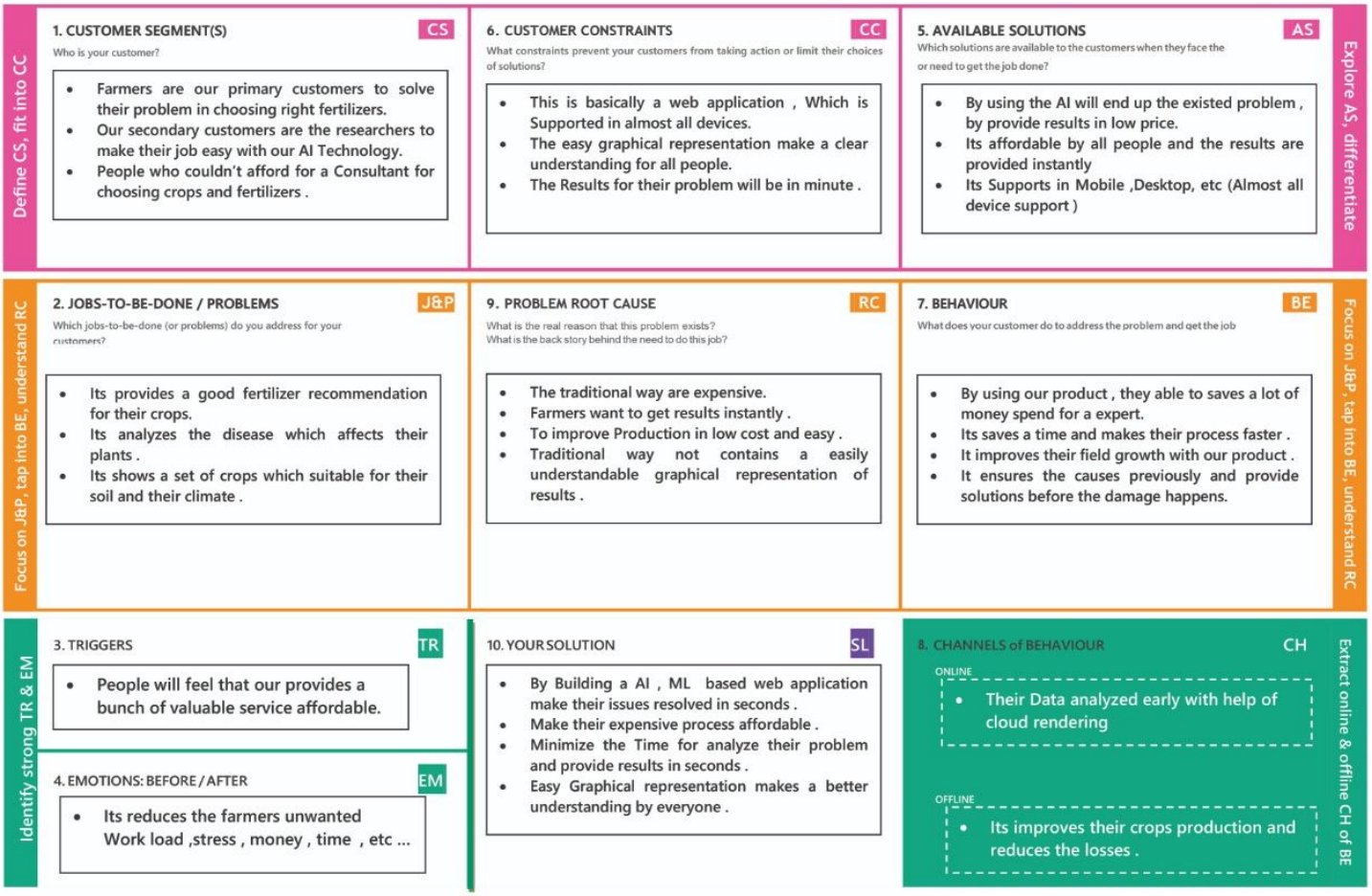


3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The majority of the farmers face the problem of planting an inappropriate plant in the land based on conventional support. The outcome for the farmer of choosing the wrong crop for the land and not the perfect fertilizer.This leads to less yield and less profit.

2.	Idea / Solution description	To overcome all these problems, a recommendation system is used.Crop and the fertilizer recommendation system is carried out by the nature of the soil like the Nitrogen,Phosphorus and Potassium.
3.	Novelty / Uniqueness	Depending upon the soil the fertilizer is recommended to the farmer.
4.	Social Impact / Customer Satisfaction	Increase productivity in agriculture and help in economic growth of our Country and the majority of the population depend on agriculture.
5.	Business Model (Revenue Model)	Analyzing the disease makes the farmer's life easier.This action adds a lot of value to the company and business in the society.
6.	Scalability of the Solution	Fertilizer Recommendation system for disease prediction provide a valuable service to the environment and society.

3.4 PROBLEM SOLUTION FIT:



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Opening Website	Fertilizer Recommendation system website should be opened.
FR-2	Uploading Data	Filling the details about the crop and soil.
FR-3	Requesting Solution	Uploaded data is compared with the pre-defined data and solution is generated
FR-4	Final Solution	The final solution is predicting the recommended fertilizer and usage of that fertilizers.

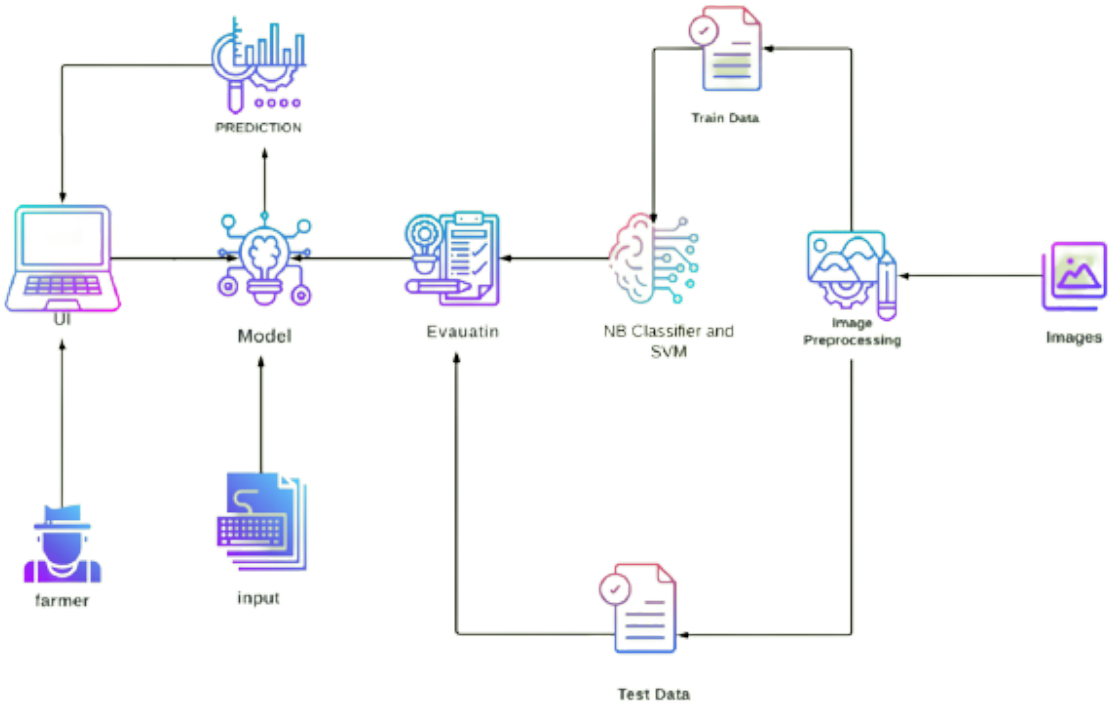
4.2 NON-FUNCTIONAL REQUIREMENTS:

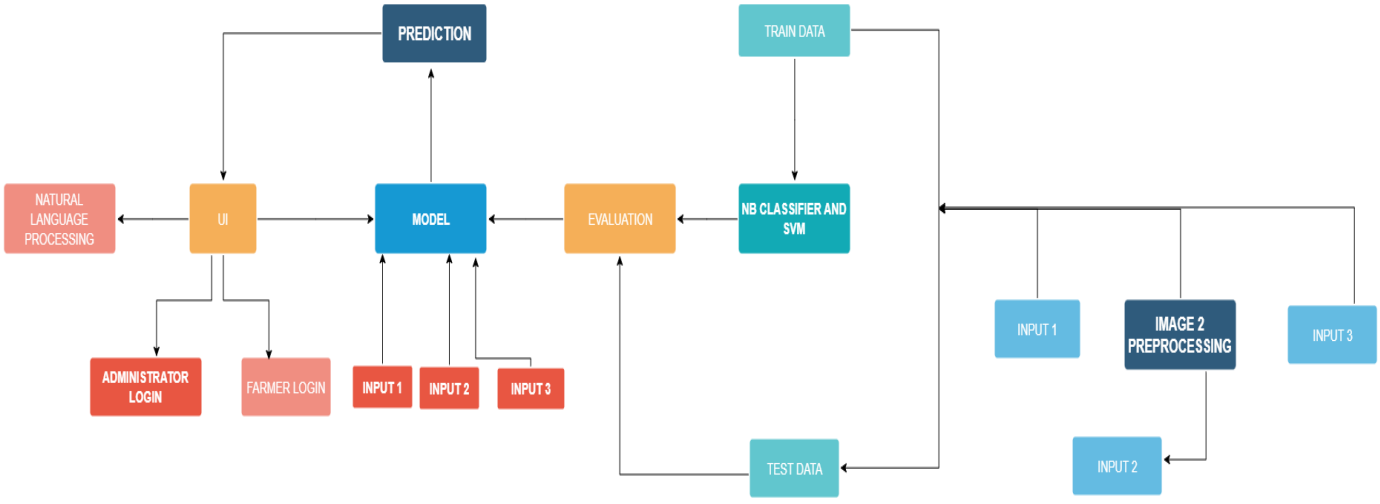
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system allows the user to perform the tasks easily and efficiently and effectively.
NFR-2	**Security**	Assuring all data inside the system or its part will be protected against malware attacks or unauthorized access.
NFR-3	**Reliability**	The website does not recover from failure quickly ,it takes time as the application is running in single server .
NFR-4	**Performance**	Response Time and Net Processing Time is Fast.
NFR-5	**Availability**	The system will be available up to 95% of the time.
NFR-6	**Scalability**	The website is scalable.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

5.2 SOLUTION & TECHNICAL ARCHITECTURE:





5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	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
	Dashboard	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
Customer (Web user)	Registration	USN-4	As a user, I can register for the application through Gmail	I can login register using my username and password	Medium	Sprint-3
	Login	USN-5	As a user, I can log into the application by entering email & password	I can use my login user credentials	High	Sprint-3
	Dashboard	USN-6	As a user , i can view the website where i can upload my images and the fertilizer should be recommended	I can access my account/dashboard	High	Sprint-4
		USN-7	As a user, the recommended fertilizer should be of high accuracy	I can access my account/dashboard	High	Sprint-4
Administrator	Login	USN-8	As a admin, i can login to the website using my login credentials	I can login to the website using my login credentials.	High	Sprint-5
	Dashboard	USN-9	As a admin, i can view the dashboard of application	I can access my dashboard	High	Sprint-5

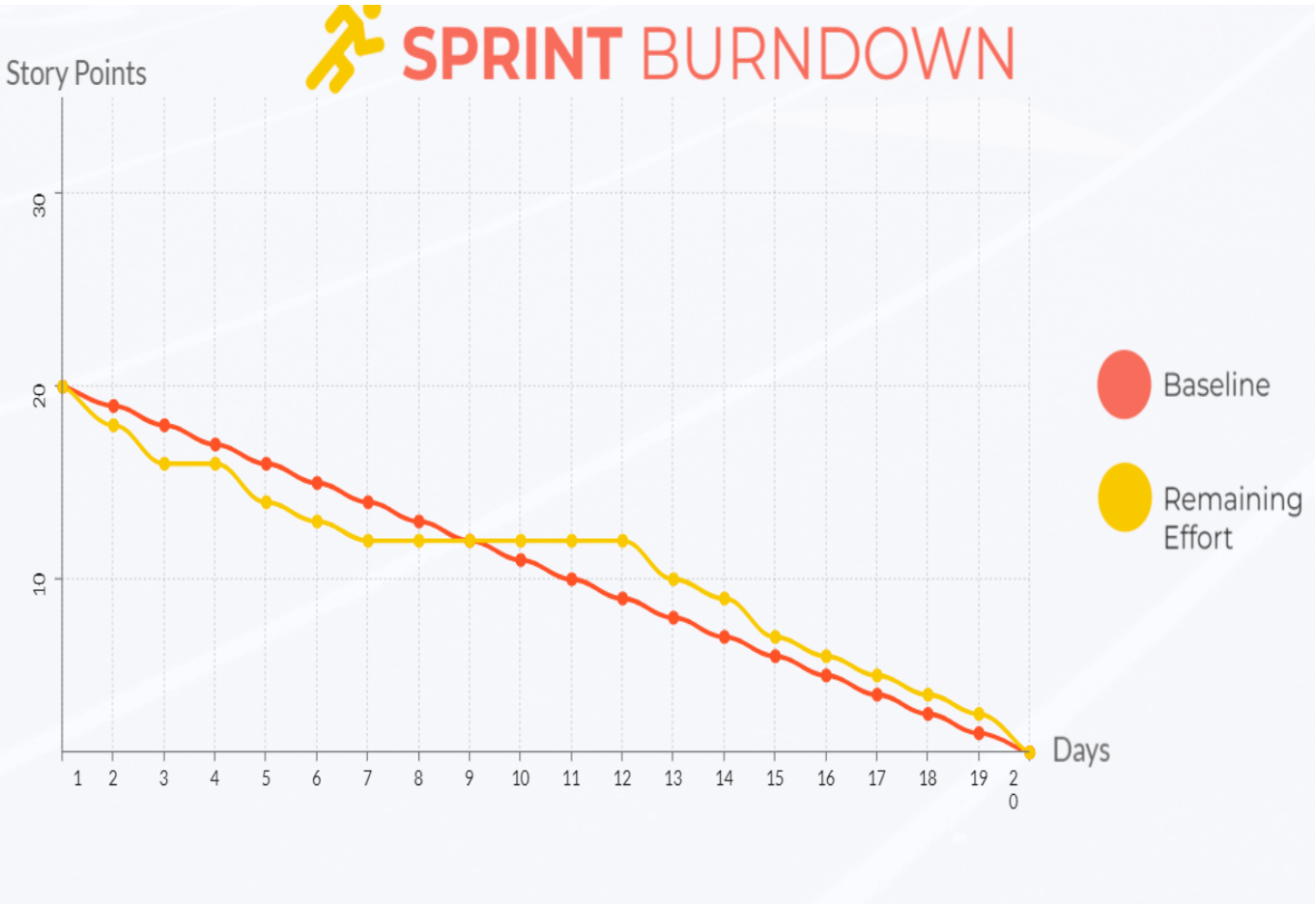
6.PROJECT PLANNING & SCHEDULING:

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image Processing.	USN-1	As a user, I can retrieve useful information about the images.	1	Low	Niranjana SR, Kirthana P, HarshaVardini G, Harini,Nowziya Banu Y
Sprint-2	Model Building for Fruit Disease Prediction.	USN-2	..As a user, I can retrieve useful information about the images.	1	Medium	Niranjana SR, Kirthana P, HarshaVardini G, Harini,Nowziya Banu Y
Sprint-2	Model Building for Vegetable Disease Prediction.	USN-3	As a user, I can able to predict vegetable disease using this model.	2	Medium	Niranjana SR, Kirthana P, HarshaVardini G, Harini,Nowziya Banu Y
Sprint-3	Application Building.	USN-4	As a user, I can see a web page for Fertilizers Recommendation System for Disease Prediction	2	High	Niranjana SR, Kirthana P, HarshaVardini G, Harini,Nowziya Banu Y
Sprint-4	Train The Model on IBM Cloud.	USN-5	As a user, I can save the information about Fertilizers and crops on IBM cloud	2	High	Niranjana SR, Kirthana P, HarshaVardini G, Harini,Nowziya Banu Y

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	26 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	10 Nov 2022



7.CODING AND SOLUTIONING

Figure.6.3. Train the Vegetable dataset

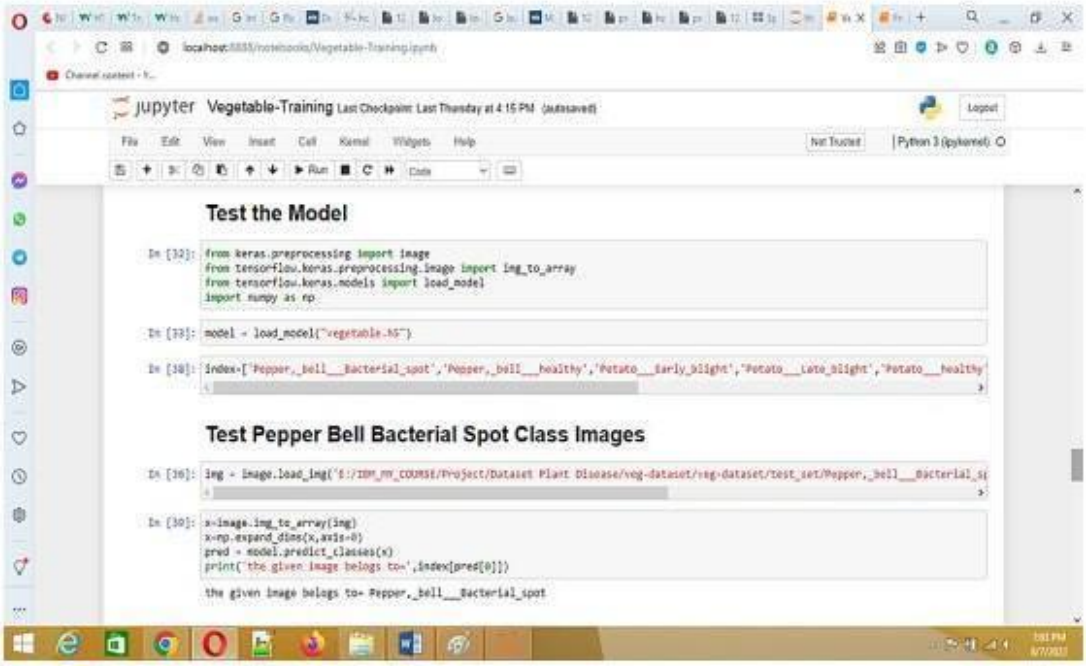


Figure.6.4. Test the Vegetable dataset

Figure.6.5. Training Fruit Dataset in IBM Cloud

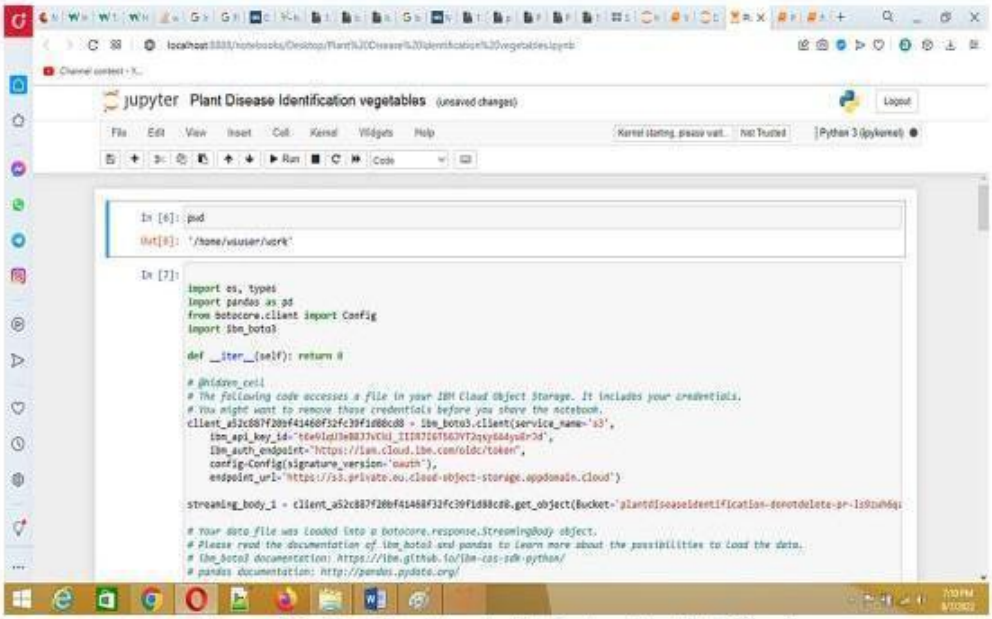
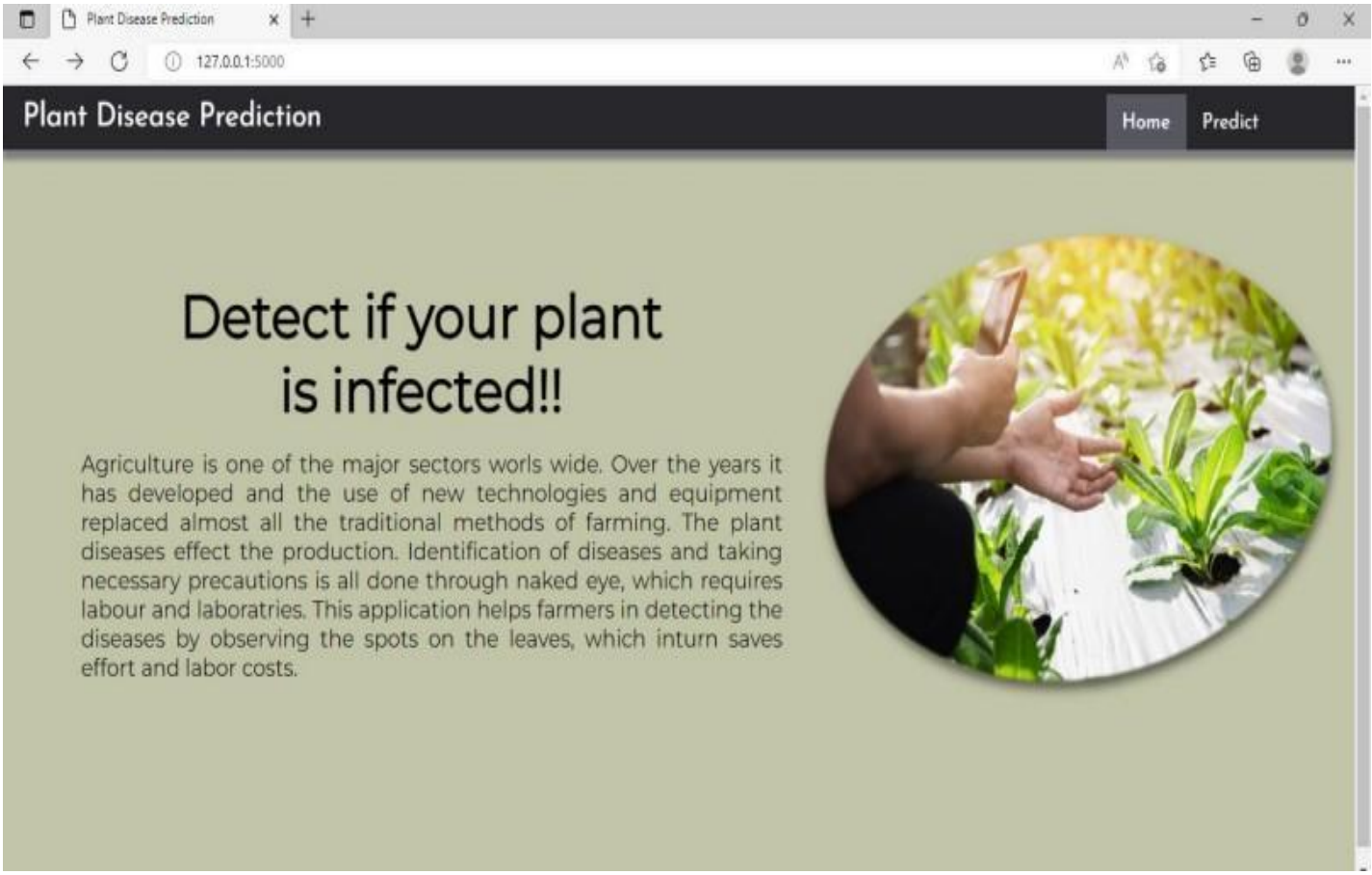


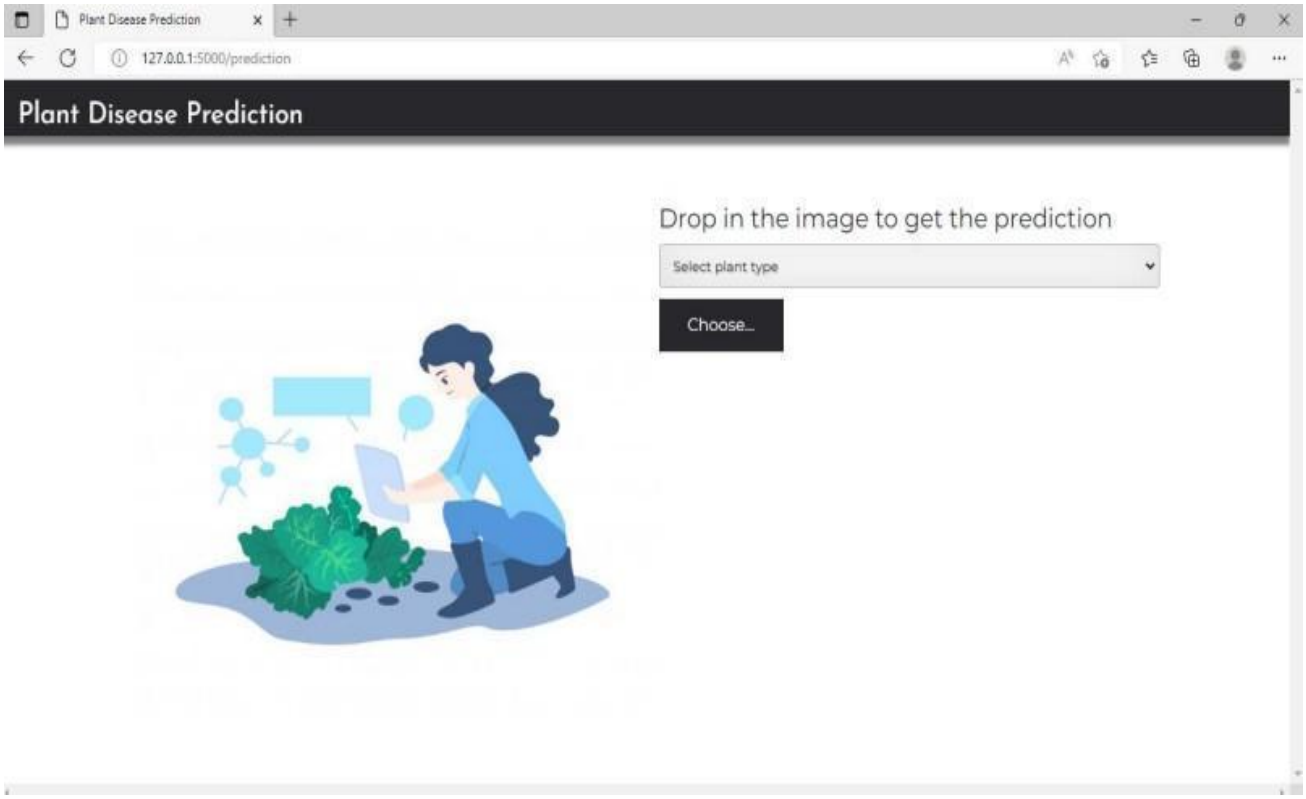
Figure.6.6. Training Vegetable Dataset in IBM Cloud

Output:

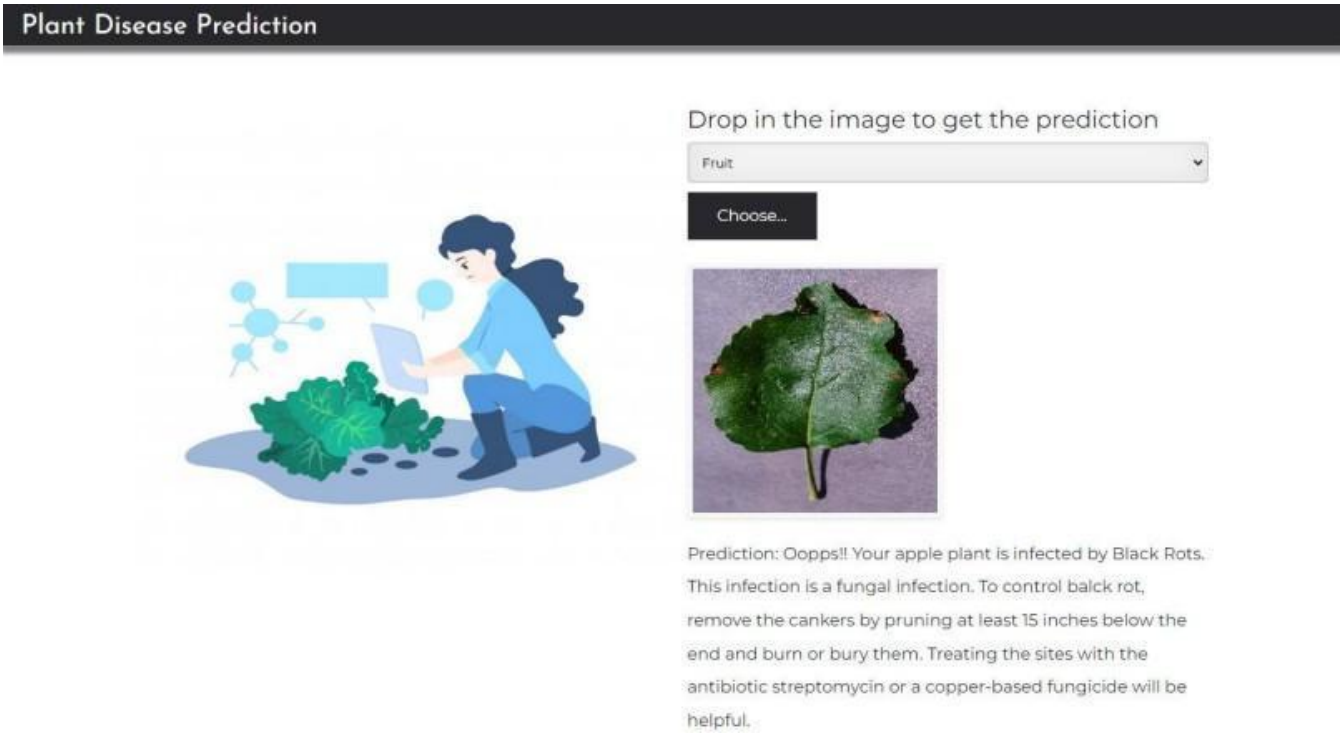
Home page:

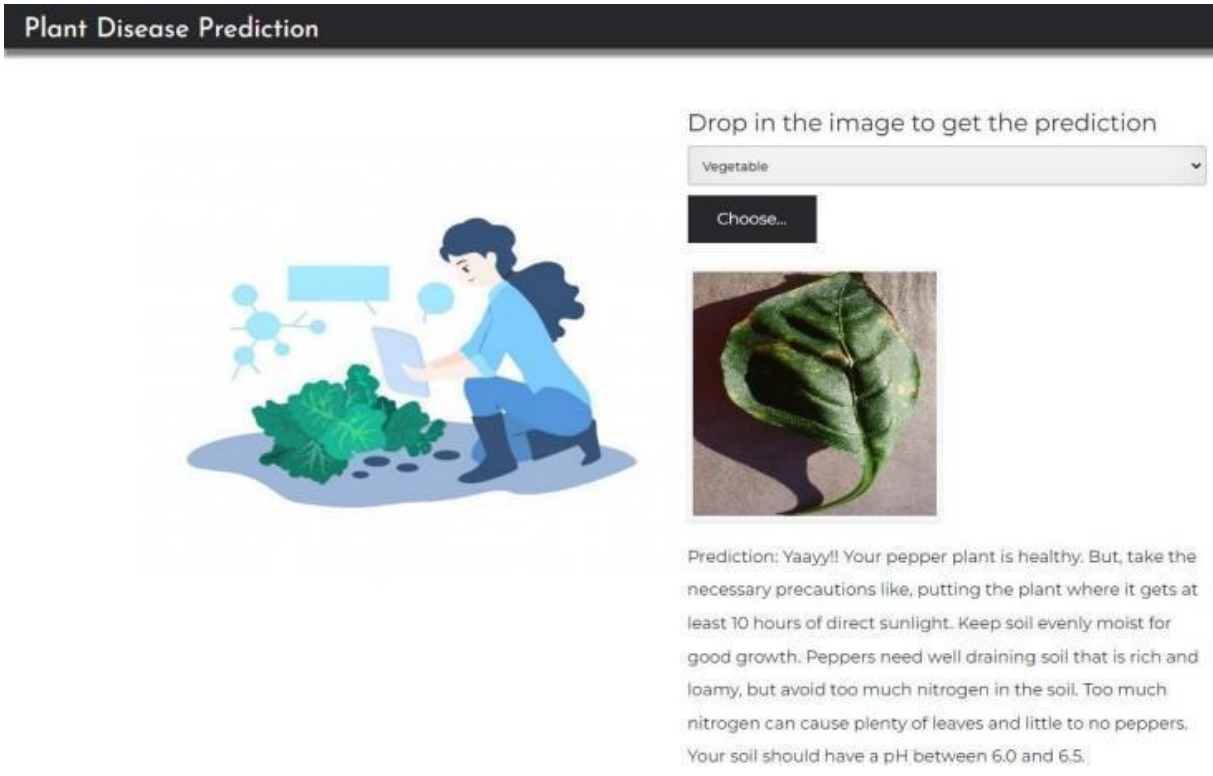


➢**Prediction Page:**



➢ **Result Page:**





8.TESTING

8.1 TEST CASES:

Section	Total Cases	Not Tested	Fail	Pass
Leaf spots	17	0	0	17
Mosaic leaf pattern	51	0	0	51
Misshapen leaves	20	0	0	20
Yellow leaves	7	0	0	7
Fruit rots	9	0	0	9
Fruit spots	4	0	0	4
Blights	2	0	0	2

8.2 USER ACCEPTANCE TESTING:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Leaf spots	10	4	2	3	19
Mosaic leaf pattern	9	6	3	6	24
Misshapen leaves	2	7	0	1	10
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	0	6
Fruit spots	5	3	1	1	10
Blights	4	5	2	1	12
Totals	44	31	13	32	119

9.ADVANTAGES & DISADVANTAGES

List of advantages

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- Images of very high can be resized within the proposed itself.

List of disadvantages

- For training and testing, the proposed model requires very high computational time.
- The neural network architecture used in this project work has high complexity

APPLICATIONS

1. The trained network model used to classify the image patterns with high accuracy.
2. The proposed model not only used for plant disease classification but also for other image pattern classification such as animal classification.
3. This project work application involves not only image classification but also for pattern recognition.

10.CONCLUSIONS

The model proposed here involves image classification of fruit datasets and vegetable datasets. The following points are observed during model testing and training:

- The accuracy of classification increased by increasing the number of epochs.
- For different batch sizes, different classification accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different while varying the size of the train and test datasets.

11.FUTURE SCOPE

The proposed model in this project work can be extended to image recognition. The entire model can be converted to application software using python to exe software. The real time image classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition, iris recognition and face recognition.

12.APPENDIX

SOURCE CODE:

Main.html

```
<!DOCTYPE html>
<html >
 <head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Plant Disease Prediction</title>
      <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
```

```
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<style>
.header {
                        top:0;
                        margin:0px;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        background-color: #28272c;
                        color: white;
                        box-shadow: 0px 8px 4px grey;
                        overflow: hidden;
                        padding-left:20px;
                        font-family: 'Josefin Sans';
                        font-size: 2vw;
                        width: 100%;
                        height:8%;
                        text-align: center;
                }
                .topnav {
  overflow: hidden;
```

```
  background-color: #333;
}
.topnav-right a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}
.topnav-right a:hover {
  background-color: #ddd;
  color: black;
}
.topnav-right a.active {
  background-color: #565961;
  color: white;
}
.topnav-right {
  float: right;
  padding-right:100px;
}
body {
```

```
  background-color:#ffffff;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
  .button {
  background-color: #28272c;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  border-radius: 12px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
  width: 100%;
```

```
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
button {
  background-color: #28272c;
  color: white;
  padding: 14px 20px;
  margin-bottom:8px;
  border: none;
  cursor: pointer;
  width: 15%;
  border-radius:4px;
}
button:hover {
  opacity: 0.8;
}
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
```

```
}
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}
img.avatar {
  width: 30%;
  border-radius: 50%;
}
.container {
  padding: 16px;
}
span.psw {
  float: right;
  padding-top: 16px;
}
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
```

```
    width: 100%;
  }
}
.home{
        margin:80px;
  width: 84%;
  height: 500px;
  padding-top:10px;
  padding-left: 30px;
}
.login{
        margin:80px;
        box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid blue;
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid blue;
```

```
}
.mySlides {display: none;}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}
/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
```

```
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
.active {
  background-color: #717171;
}
/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}
@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}
@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
```

```
 }
 /* On smaller screens, decrease text size */
 @media only screen and (max-width: 300px) {
   .text {font-size: 11px}
 }
 </style>
 </head>
<body style="font-family:'Times New Roman', Times, serif;background-color:#C2C5A8;">
 <div class="header">
        <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>
   <div class="topnav-right"style="padding-top:0.5%;">
     <a class="active" href="{{ url_for('home')}}">Home</a>
     <a href="{{ url_for('prediction')}}">Predict</a>
   </div>
 </div>
 <div style="background-color:#ffffff;">
 <div style="width:60%;float:left;">
 <div
style="font-size:50px;font-family:Montserrat;padding-left:20px;text-align:center;padding-top:10%;">
 <b>Detect if your plant<br> is infected!!</b></div><br>
 <div
style="font-size:20px;font-family:Montserrat;padding-left:70px;padding-right:30px;text-
```

align:justify;">Agriculture is one of the major sectors worls wide. Over the **years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming. The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratries. This application helps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labor costs.

**

** **

<img src="{{url_for('static',filename='images/12456.png')}}" style="max-height:100%;max-width:100%;">

<script>

var slideIndex = 0;

showSlides();

function showSlides() {

  var i;

  var slides = document.getElementsByClassName("mySlides");

  var dots = document.getElementsByClassName("dot");

  for (i = 0; i < slides.length; i++) {

```
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

Fertilizers.py:

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
```

```
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session
app = Flask(__name__)
global sess
global graph
graph=tf.compat.v1.get_default_graph()
model = load_model("C:\\Users\\VISHWANTH BAVIREDDY\\Guided Project Buildathon\\Guided project\\fruit.h5")
model1=load_model("C:\\Users\\VISHWANTH BAVIREDDY\\Guided Project Buildathon\\Guided project\\vegetable.h5")
@app.route('/')
def home():
    return render_template('home.html')
@app.route('/prediction')
def prediction():
    return render_template('predict.html')
@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        f = request.files['image']
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'Dataset Plant Disease', secure_filename(f.filename))
        f.save(file_path)
```

```
    img = image.load_img(file_path, target_size=(128, 128))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    plant=request.form['plant']
    print(plant)
    if(plant=="vegetable"):
        preds = model.predict(x)
        preds = np.argmax(preds)
        print(preds)
        df=pd.read_excel('precautions - veg.xlsx')
        print(df.iloc[preds]['caution'])
    else:
        preds = model1.predict(x)
        preds = np.argmax(preds)
        df=pd.read_excel('precautions - fruits.xlsx')
        print(df.iloc[preds]['caution'])
    return df.iloc[preds]['caution']
```

GITHUB ID:

https://github.com/IBM-EPBL/IBM-Project-9385-1658998708