Date	03 November 2022
Team ID	PNT2022TMID43438
Project Name	Project – Fertilizer Recommendation System For Disease Prediction
Team Members	Mounica. S, A.Ananthaganesh, R.Gowtham, P.Varshini

PROJECT REPORT

1.Introduction

Fertilizer Recommendation system for disease Prediction is a simple ML and DL based website which recommends the best crop to grow, fertilizers to use and the diseases caught by your crops.

1.1 Project Overview:

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

1.2. Purpose:

Fertilizer Recommendation system is used in agriculture area so that efficient recommendation will be provided to the customers for selecting an appropriate fertilizer for the crops so that the production will increase and environment will be less affected.

Fertilizers are additional substances supplied to the crops to increase their productivity. These are used by the farmers daily to increase the crop yield. These fertilizers contain essential nutrients required by the plants, including nitrogen, potassium, and phosphorus.

2.Literature survey

A literature survey entails a thorough and comprehensive examination of all forms of existing literature in addition to additional sources, notably dissertation, to identify quite so many articles as necessary that are applicable to a certain topic. Predicting nutrients for plant diseases is

extremely essential in agriculture. It aids in boosting net product, improved planning, and profit maximization. Thus many research articles were reviewed connected to our project issue in order to attain better outcomes.

2.1 Existing problem:

- **"Cloud Based Automated Irrigation and Plant Leaf Disease Detection System Using an Android Application."**

Authors: Ranjith, Saheer Anas, Ibrahim Badhusha, O.T. Zaheema , K Faseela,Minnuja Shelly.

Publication: International Conference on Electronics, Communication and AerospaceTechnology, ICECA 2017.

The major goal of this study is to provide imaging techniques and classification techniques for recognising and categorising leaf diseases. The leaf picture is first pre- processed before being further processed. K-Means Clustering is employed to segment the pictures, and the system retrieves the GLCM features first from images that include illness detection. Disease categorization via the SVM classifier. The system predicts and identifies illnesses on citrus leaves with a maximum of 90% accuracy; however, the system can only identify diseases on citrus leaves.

Algorithm used: SVM, Gray-Level Co-Occurrence Matrix (GLCM) features, K-Means Clustering. Advantages:It is simple and cost effective system for plant leaf disease detection. Disadvantages:Any H/w failures may affect the system performance. The current paper proposesan android application for irrigation and plant leaf disease detec tion with cloud and IoT. For monitoring irrigation system they use soil moisture and temper ature sensor and sensor data send to the cloud. The user can also detect the plant leaf disease.

- **"Crop Recommendation System for Precision Agriculture."**

Authors: S.Pudumalar, E.Ramanujam, R.Harine Rajashreen, C.Kavyan, T.Kiruthikan, J.Nishan. Publication: IEEE Eighth International Conference on Advanced Computing(ICoAC) 2016.

The method reported in the study incorporates the Majority Voting Strategy, an ensemble technique that harnesses the power of many models to increase prediction accuracy. For ensemble, numerous algorithms are used, such that even though one technique makes an incorrect prediction, the other ways are expected to forecast correctly, and the accuracy rate is correct since the majority voting mechanism is used. If-then rules are the fundamental components used in the prediction phase. The ensemble model has an 88% accuracy. Algorithm used: Random tree forest,KNN,Naive Bayes.

- **"Use of Data Mining in Crop Yield Prediction."**

Authors: Shruti Mishra, Priyanka Paygude, Snehal Chaudhary, Sonali Idate. Publication: Proceedings of the Second International Conference on Inventive Systems and Control (ICISC) 2018.

The data set employed in this work was obtained from Kaggle.com. Theauthor used the WEKA tool to analyse the data for the algorithms LWL, J48, LAD Tree, and IBK. Specificity, sensitivity, accuracy were used to calculate accuracy. To get the correctly detected cases for each classifier, a confusion matrix was employed. The conclusion was that trimming can improve accuracy. Algorithm used: J48,LAD tree,LWL,IBK algorithm. A St

- **"A Study on Various Data Mining Techniques for Crop Yield Prediction."**

Authors: Yogesh Gandge, Sandhya. Publication: International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT) 2017.

Multiple Linear Regression was shown to have an efficiency of 90-95% for rice productivity in this study. For the soybean crop, a decision tree with the ID3 algorithm was analyzed, and suggestions were created. The third approach was SVM, which was employed on all crops and provided high accuracy while requiring little processing effort. On corn data, a neural network was utilised to attain 95% accuracy. The result was that further work is needed to increase the algorithms' accuracy.

Algorithm used: Multiple Linear Regression,Decision Tree, SVM, Neural Networks.

- **"Fertilizers Recommendation System For Disease Prediction In Tree Leave."**

Authors: R. Neela, P.Nithya. Publication: International Journal Of Scientific & Technology Research Volume 8,Issue 11, November 2019.

By recommending the best crops, this proposal aids in our ability to estimate crop productivity. In order to determine what crop should be put in the field to enhance productivity, it also concentrates on soil types. Soil types are crucial for crop yield. Information about the soil can be acquired by factoring in the weather information from the previous year. It enables us to foresee which crops might thrive in a specific climate. Algorithms for prediction assist us in categorizing the data according to the disease, and information extrapolated out from classifier is utilized toforecast soil and crop. However, this approach is unable to anticipate exact outcomes because of the fluctuating climatic conditions.

Algorithm used: SVM Advantages:It allows us to predict which crops would be appropriate for a given climate. Using the weather and disease related data sets, the crop quality can also be improved. Pre diction algorithms help us to classify the data based on the disease, and data extracted from the classifier is used to predict soil and crop.

Disadvantages:Due to the changing climatic conditions, accurate results cannot be predicted by this system.

- **"Design of Intelligent Agriculture Management Information System Based onIOT."**

Authors: Duan Yane. Publication: IEEE,4th, Fourth International reference on Intelligent Computation Technology and Automation, 2011.

The current paper suggests an Android mobile application for cloud and IoT- based irrigation and plant detecting leaf diseases. They utilize soil moisture content, temperature sensors, and sensor data sent to the cloud to monitor irrigation systems. K-means clustering is used to extract features.The system's simplicity and affordability make it a good choice for detecting plant leaf diseases. However, any hardware or design flaws could impair the system's performance.

Algorithm used: SVM 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

2.2 References:

- 2018 the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis, **"Big Data Analysis Technology Application in Agricultural Intelligence Decision System",** Jichun Zhao; Jian-xin Guo.
- Cloud Based Automated Irrigation And Plant Leaf Disease Detection System Using An Android Application. International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017.
- 2016 IEEE Eighth International Conference on Advanced Computing (ICoAC), **"Crop Recommendation System for Precision Agriculture"**, S.Pudumalar, E.Ramanujam, R.Harine Rajashreen, C.Kavyan, T.Kiruthikan, J.Nishan.
- Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018), **"Use of Data Mining in Crop Yield Prediction**", Shruti Mishra, Priyanka Paygude, Snehal Chaudhary, Sonali Idate.
- 2017 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), **"A Study on Various Data Mining Techniques for Crop Yield Prediction"**, Yogesh Gandge, Sandhya.
- R. Neela, P.Nithya Fertilizers Recommendation System For Disease Prediction In Tree Leave International Journal of Scientific & Technology Research Volume 8, Issue 11, November 2019.
- Duan Yane, Design of Intelligent Agriculture Management Information System Based on IOT, IEEE,4th, Fourth International reference on Intelligent Computation Technology and Automation, 2011.

2.3 problem statement definition:

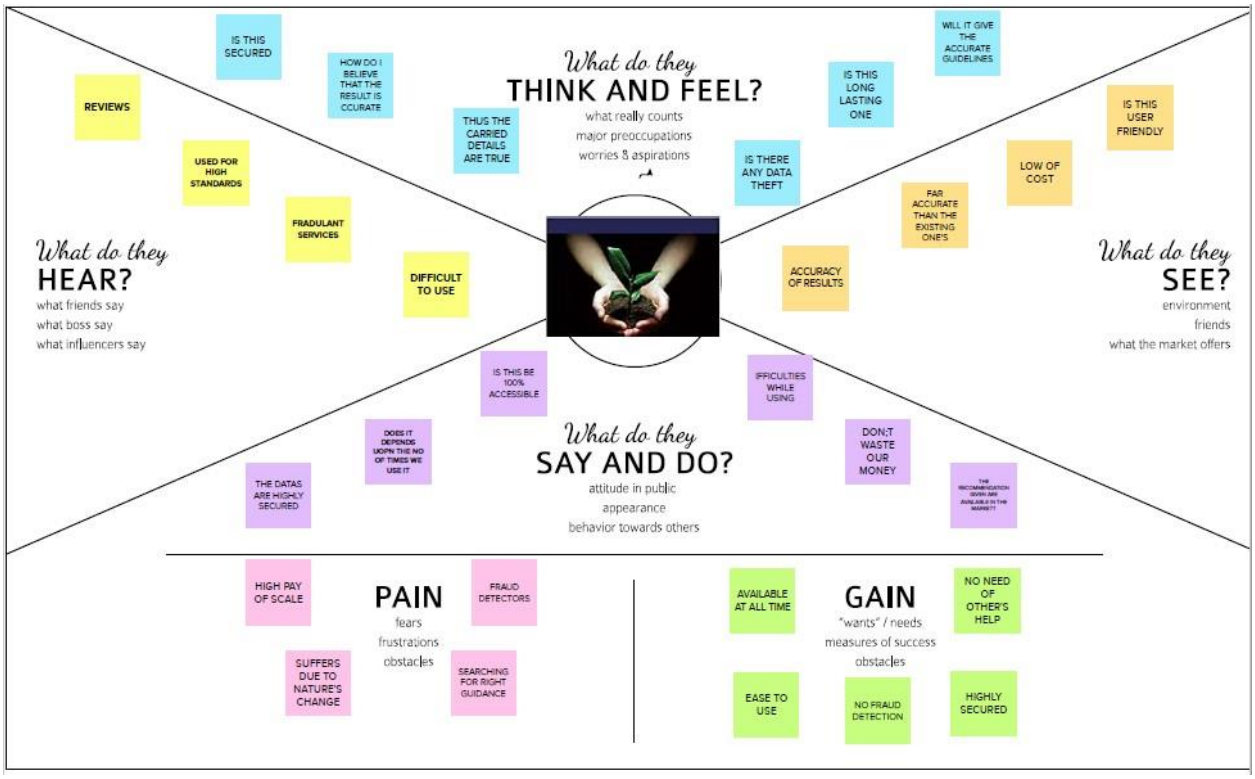
Mano is a 35 years old man. He has a farming land where he cultivates his crops and he make sales on market for his daily needs. He also does Agriculture for 15 years. During this time, he has faced a problem in choosing a correct pesticide / fertilizer for his infected crops. He has a confusion in choosing Fertilizers and controlling of disease in plants.

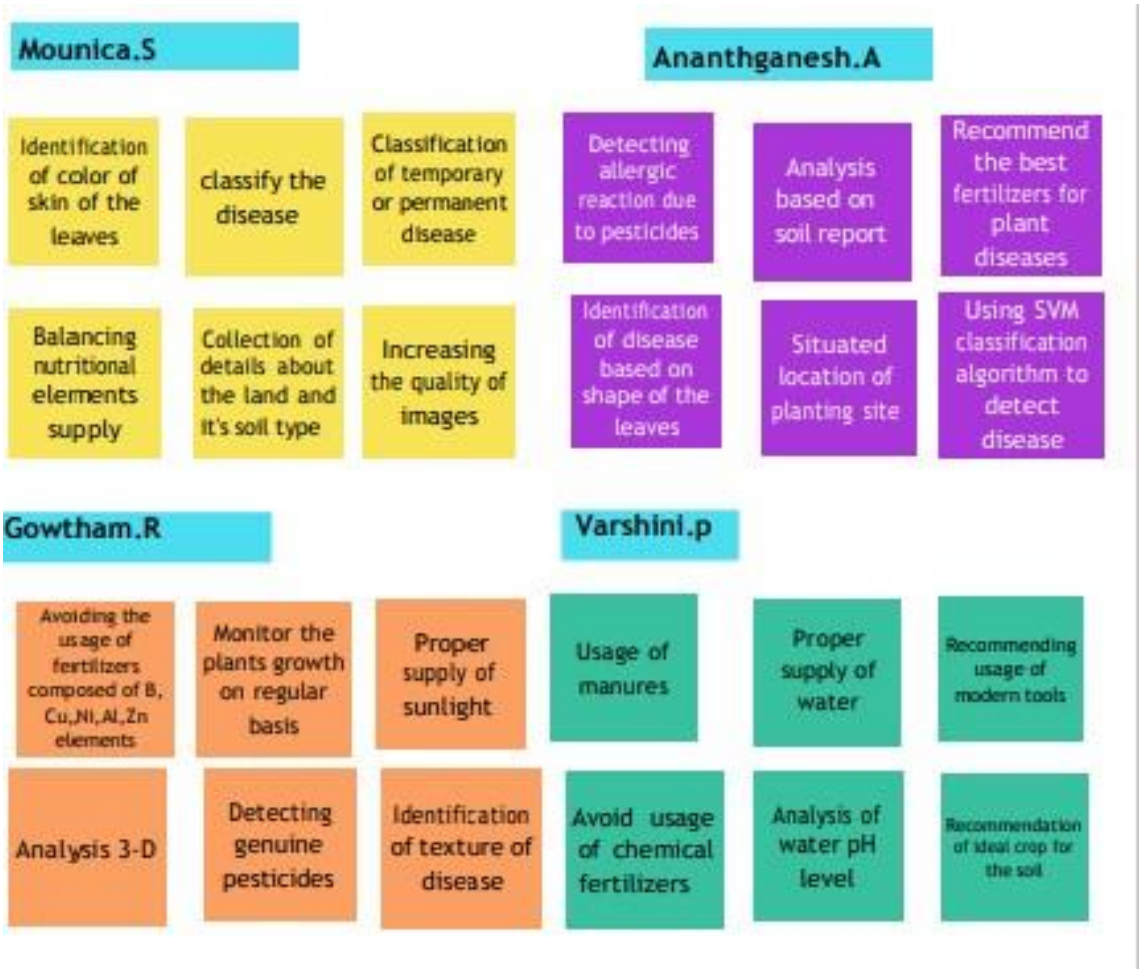
- Mano Needs to know the best recommendation fertilizers for his infected plants.
- He Need to know the result as soon as possible.

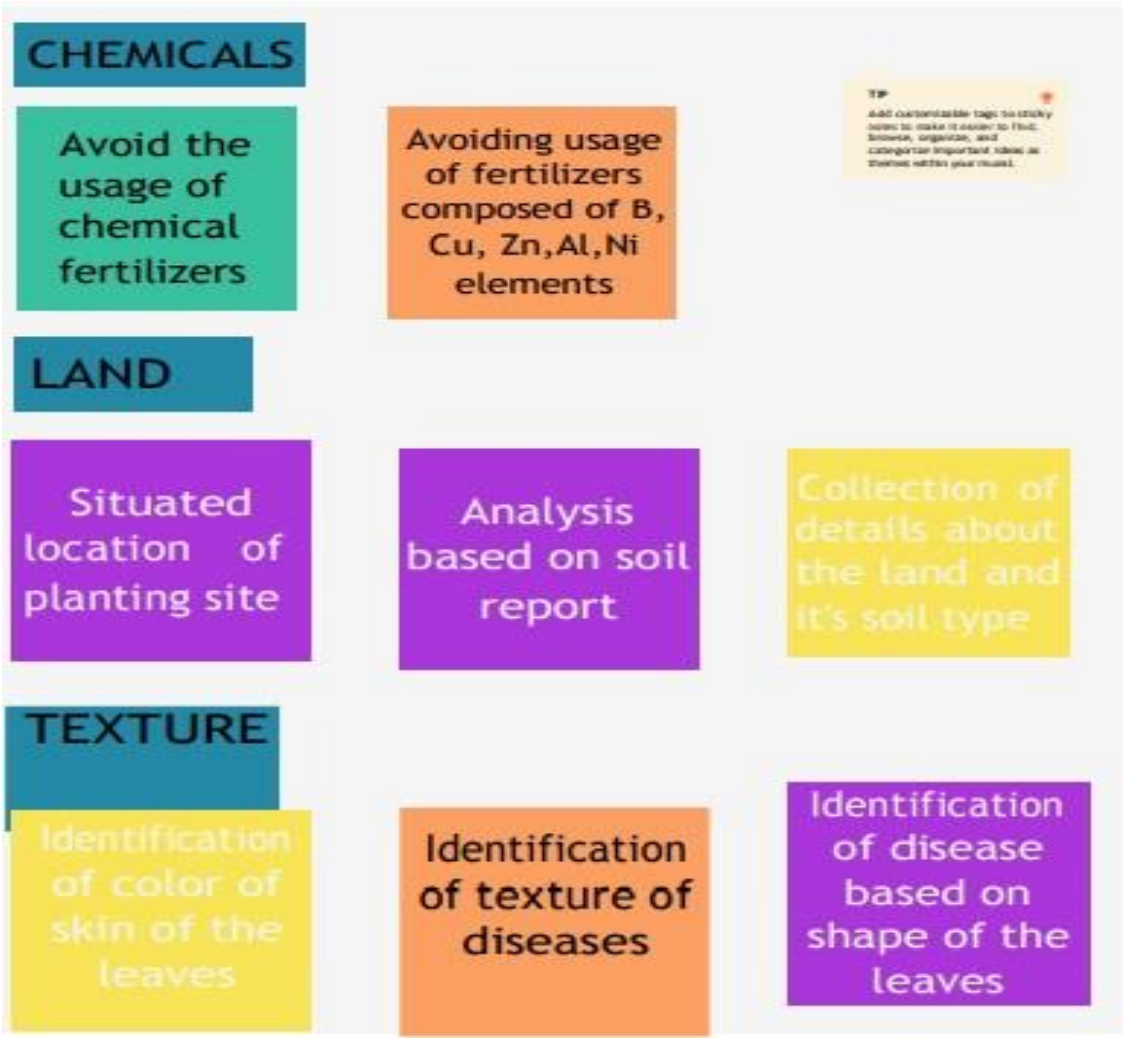
I am Mano, a 35 Years old man Doing agriculture for the past 15 years for my daily income. I'm trying to Cultivate Crop without getting affected by the infected plants. But, I am unable to find a perfect and correct fertilizer for a crop which makes the plant to get free from infected disease. Because Plants might get affected due to some infected viruses and I don't know the correct fertilizer to provide to those crops. Which makes me feel My crops will get infected due to these viruses and I couldn't able to provide an appropriate fertilizer for my crops which makes me sad and it causes loss of money and I can't able to sell on market so I will lose my daily earnings too.

3. Ideation &Proposed Solution

3.1 Empathy Map Canvas:




Mounica.S
Identification of color of skin of the leaves
classify the disease
Classification of temporary or permanent disease
Balancing nutritional elements supply
Collection of details about the land and it's soil type
Increasing the quality of images
Ananthganesh.A
Detecting allergic reaction due to pesticides
Analysis based on soil report
Recommend the best fertilizers for plant diseases
Identification of disease based on shape of the leaves
Situated location of planting site
Using SVM classification algorithm to detect disease
Gowtham.R
Avoiding the usage of fertilizers composed of B, Cu,Ni,Al,Zn elements
Monitor the plants growth on regular basis
Proper supply of sunlight
Analysis 3-D
Detecting genuine pesticides
Identification of texture of disease
Varshini.p
Usage of manures
Proper supply of water
Recommending usage of modern tools
Avoid usage of chemical fertilizers
Analysis of water pH level
Recommendation of ideal crop for the soil



3.3 Proposal solution:

Problem Statement (Problem to be solved) :

Diseases on plant reduced crop productivity and serious danger to farmers' livelihoods and food security. Thus, to ensure that crops are produced in large quantities and at the highest possible standards, it is crucial to identify plant illnesses early and accurately and to use the appropriate nutrients.

Idea / Solution description :

To create a web application that will help farmers foresee crop infestations. New technologies like machine learning and deep learning algorithms can forecast it. Along with forecasting, we also intended to advise on the appropriate pesticide to use and the recommended dosage in order to save the crop from degeneration. The CNN algorithm is

then used to form a neural network for predicting crop disease after the train and test picture datasets have been preprocessed. To provide the farmers with an interface, a web application built with Flask is developed.

Novelty / Uniqueness:

The software advises both organic and inorganic fertilizers , and farmers can place online orders for the fertilisers that are readily accessible.

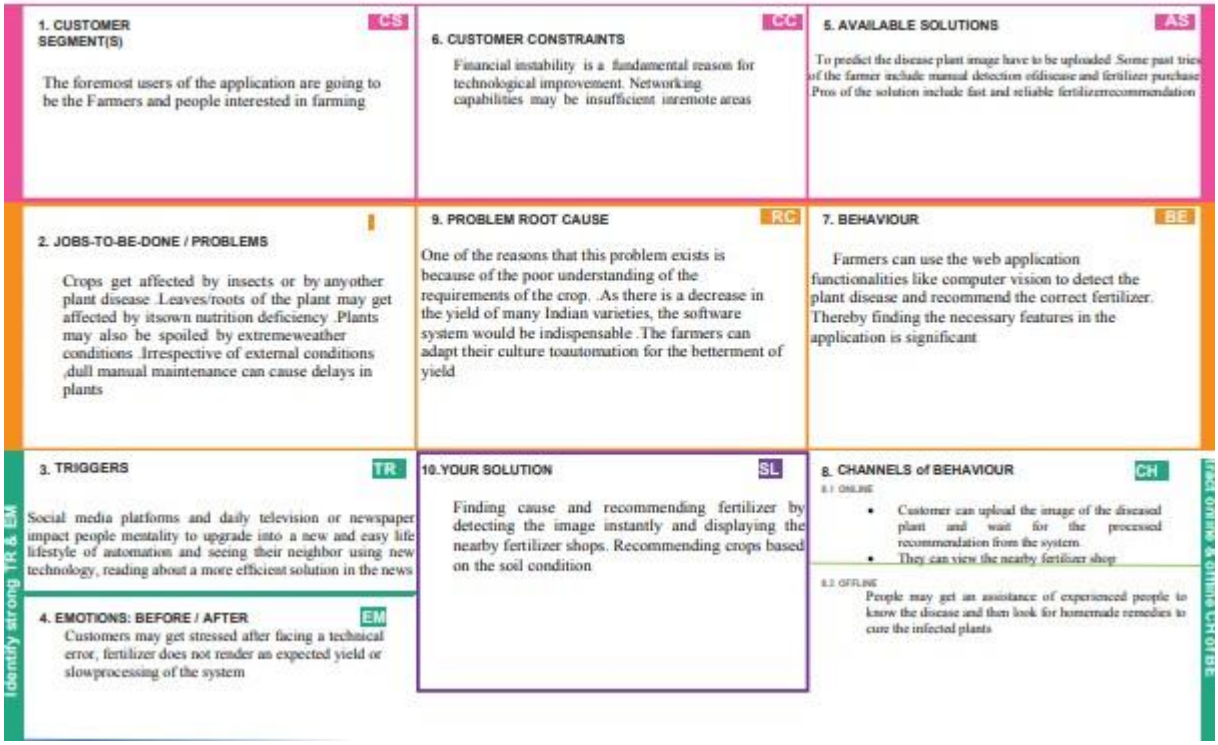
Business Model (Revenue Model):

Helps farmers produce food goods in a good way and cuts down on production losses earlier. Crop yield, crop efficiency, and agricultural product output will all rise with the suggested approach. Agriculture output will expand significantly, and profit will be increased.

Scalability of the Solution:

The proposed structure for precision agriculture enables the application of a flexible methodology that may be modified for various types of crops. Based on the taught data, the software will anticipate an accurate answer.

3.4 Problem solution fit:


1. CUSTOMER SEGMENT(S)
CS
The foremost users of the application are going to be the Farmers and people interested in farming
6. CUSTOMER CONSTRAINTS
CC
Financial instability is a fundamental reason for technological improvement. Networking capabilities may be insufficient inremote areas
5. AVAILABLE SOLUTIONS
AS
To predict the disease plant image have to be uploaded .Some past tries of the farmer include manual detection ofdisease and fertilizer purchase .Pros of the solution include fast and reliable fertilizerrecommendation
2. JOBS-TO-BE-DONE / PROBLEMS
Crops get affected by insects or by anyother plant disease .Leaves/roots of the plant may get affected by itsown nutrition deficiency .Plants may also be spoiled by extremeweather conditions .Irrespective of external conditions ,dull manual maintenance can cause delays in plants
9. PROBLEM ROOT CAUSE
RC
One of the reasons that this problem exists is because of the poor understanding of the requirements of the crop. .As there is a decrease in the yield of many Indian varieties, the software system would be indispensable .The farmers can adapt their culture toautomation for the betterment of yield
7. BEHAVIOUR
BE
Farmers can use the web application functionalities like computer vision to detect the plant disease and recommend the correct fertilizer. Thereby finding the necessary features in the application is significant
3. TRIGGERS
TR
Social media platforms and daily television or newspaper impact people mentality to upgrade into a new and easy life lifestyle of automation and seeing their neighbor using new technology, reading about a more efficient solution in the news
4. EMOTIONS: BEFORE / AFTER
EM
Customers may get stressed after facing a technical error, fertilizer does not render an expected yield or slowprocessing of the system
10.YOUR SOLUTION
SL
Finding cause and recommending fertilizer by detecting the image instantly and displaying the nearby fertilizer shops. Recommending crops based on the soil condition
8. CHANNELS of BEHAVIOUR
CH
8.1 ONLINE
Customer can upload the image of the diseased plant and wait for the processed recommendation from the system.
They can view the nearby fertilizer shop
8.2 OFFLINE
People may get an assistance of experienced people to know the disease and then look for homemade remedies to cure the infected plants
Identify strong TR & EM
Extract online & offline CH of BE

4.Requirement Analysis

4.1 Functional requirement :

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	Image Capture	Capture image of leaf Check the leaf is captured under given parameters
FR-3	Image Processing	Upload the leaf image Start detection
FR-4	Leaf Prediction	Identify the parameter to be considered for the identification of diseases.
FR-5	Image Description	Show the prescribed fertilizer to be used for un healthy leaf
FR-6	Providing Dataset	Training datasets Testing datasets
FR-7	Adding Datasets	Fruit dataset and vegetable dataset
FR-8	Updated Native Language options	Languages can be changed according to the user wish

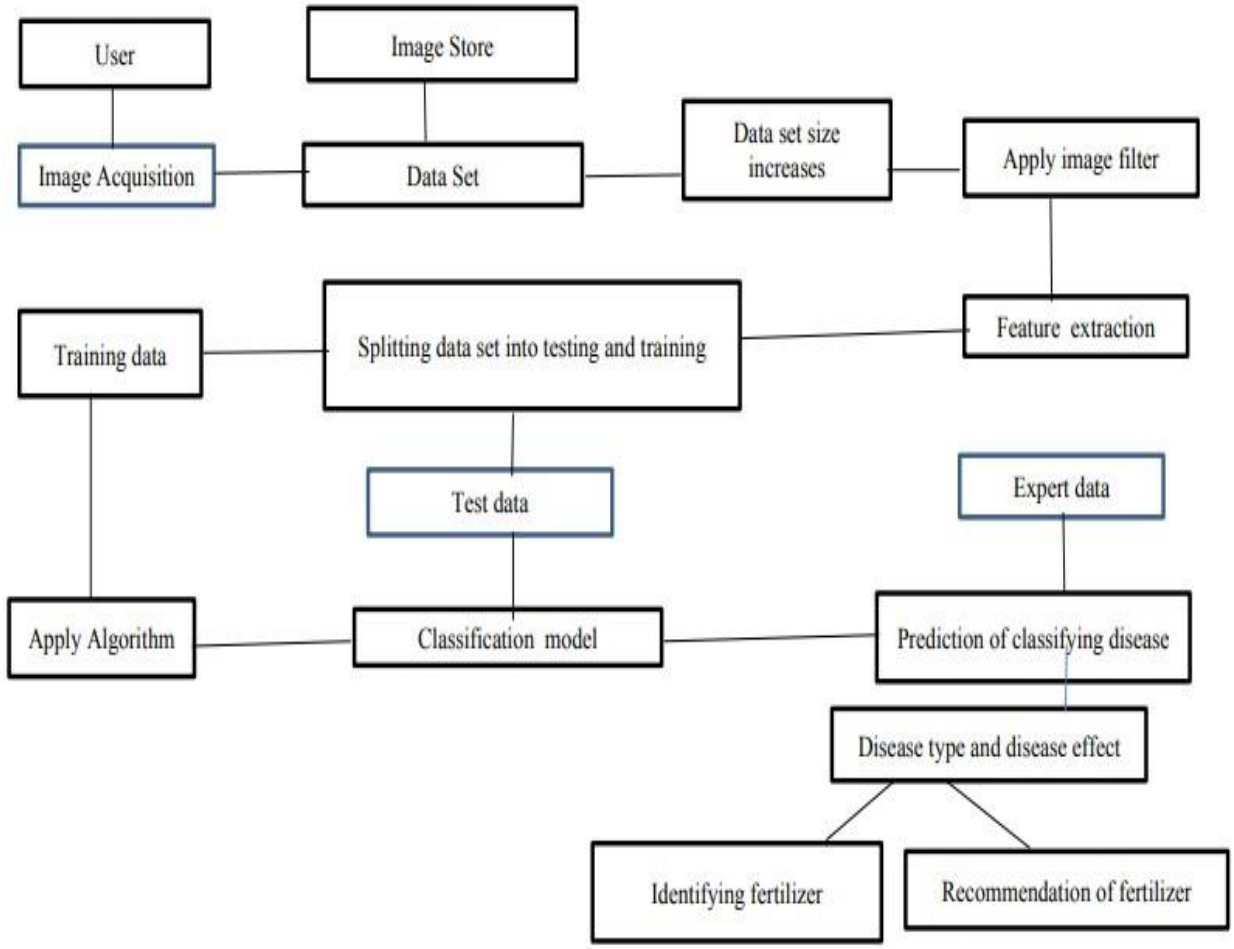
4.2 Non Functional Requirement:

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

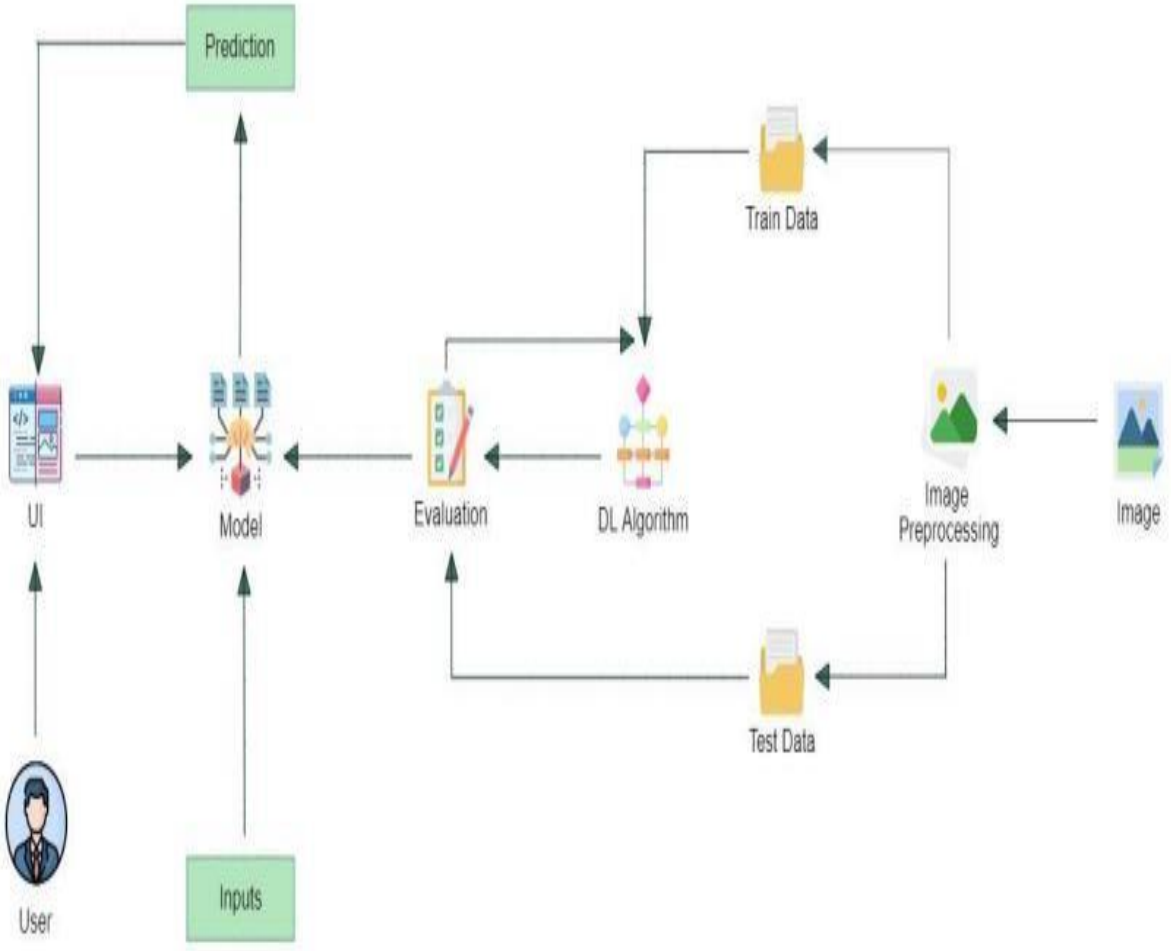
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Leaf datasets can be used for detection of all kind of leaf's Datasets can be reusable Data sets can be prepared according to the leaf
NFR-2	**Security**	User information and leaf data are secured The algorithms used are more secure
NFR-3	**Reliability**	The leaf quality is more The datasets and image capturing performs consistently well
NFR-4	**Performance**	Leaf problem defines once the leaf is detected Performs well according to the quality of leaf provides certain cure to it.
NFR-5	**Availability**	Quality of leaf will be used again for detection Available and easy access of

5.Project Design

5.1 Data Flow diagram :



5.2 Solution &Technical Architecture:



5.3Users 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	Confirmation through email	High	Sprint-1
	Login	USN-3	As a user, I can register for the application through Facebook	Login detail were made is successful.	High	Sprint-1
	Dashboard	USN-4	As a user, I can register for the application through Gmail		Low	Sprint-2
Customer (Web user)	Registration	USN-1	As a user, I can log into the application by entering email & password and confirmation through mail	They can access the account details and dashboard.	High	Sprint-1
Customer Care Executive	Login	USN-1	Login into the executive portal to help the user	Help in accessing the moment and the access	High	Sprint-1
Administrator	Help dashboard	USN-2	Can provide the necessary details of help through desired way like mobile, email, phone and SMS		Medium	
	User account control	USN-1	The person or administrator who is responsible for the website control and other management activities.	Provides support to the team or the individual	High	Sprint-1

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	S.Mounica, P.Varshini
Sprint-2	Model Building for Fruit Disease Prediction.	USN-2	As a user, I can able to predict fruit disease using this model.	1	Medium	S.Mounica, P.Varshini
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	S.Mounica, P.Varshini
Sprint-3	Application Building.	USN-4	As a user, I can see a web page for Fertilizers Recommendation System for Disease Prediction	2	High	S.Mounica, A.Ananthaganesh, R.Gowtham, P.Varshini
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	S.Mounica, P.Varshini

6.2 Sprint Delivery Schedule:

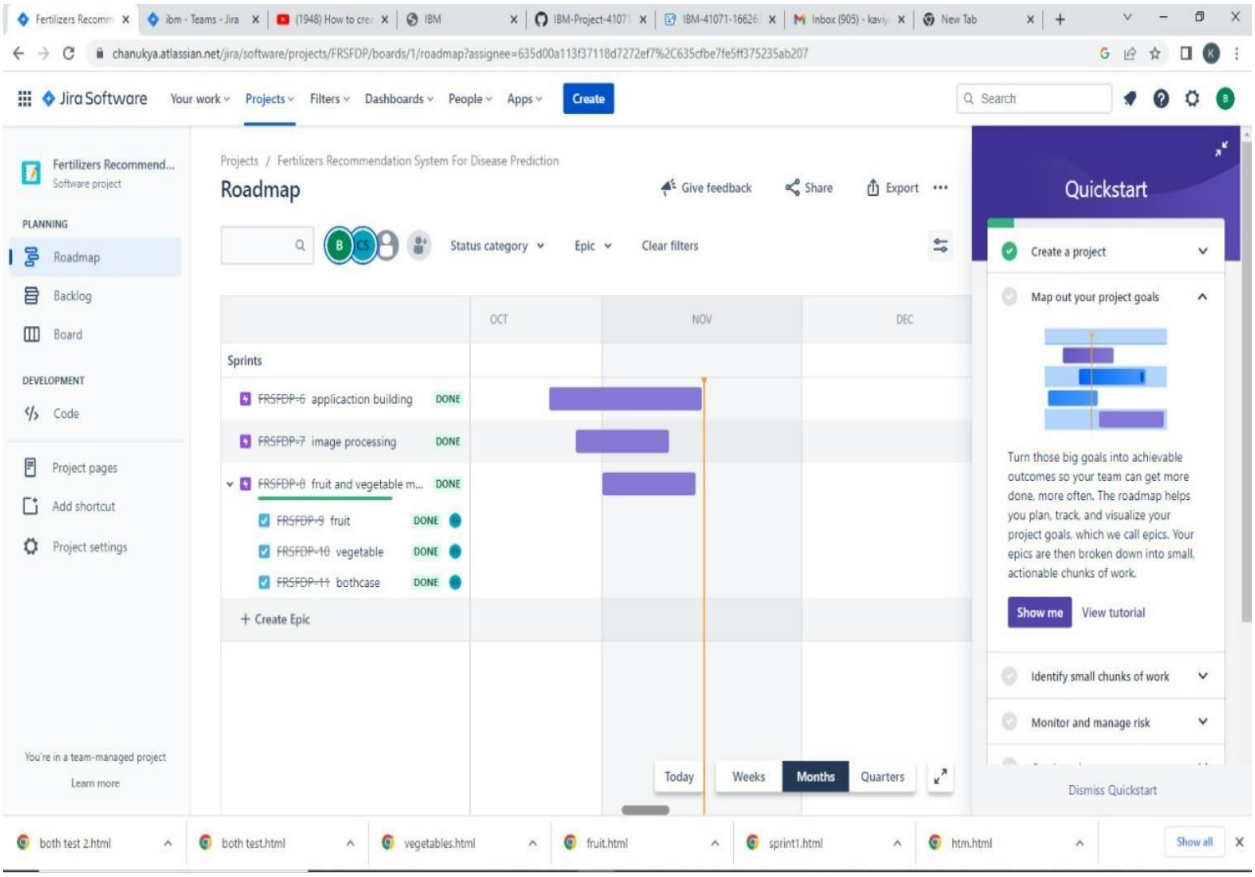
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	26 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	10 Nov 2022

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)

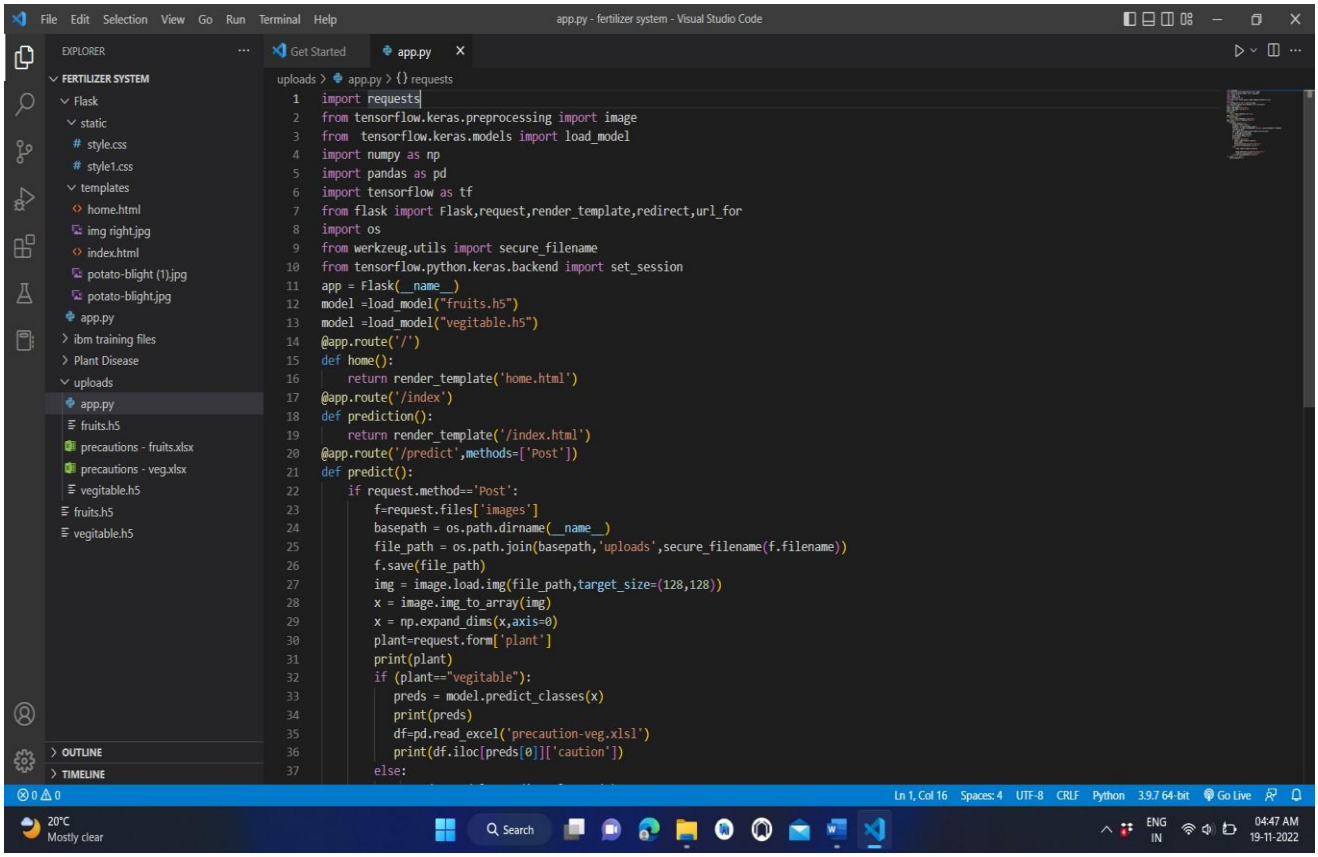
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3Reports From JIRA:



7.Coding And Solutioning

7.1 Feature 1 :



PYTHON CODE FOR FLASK:

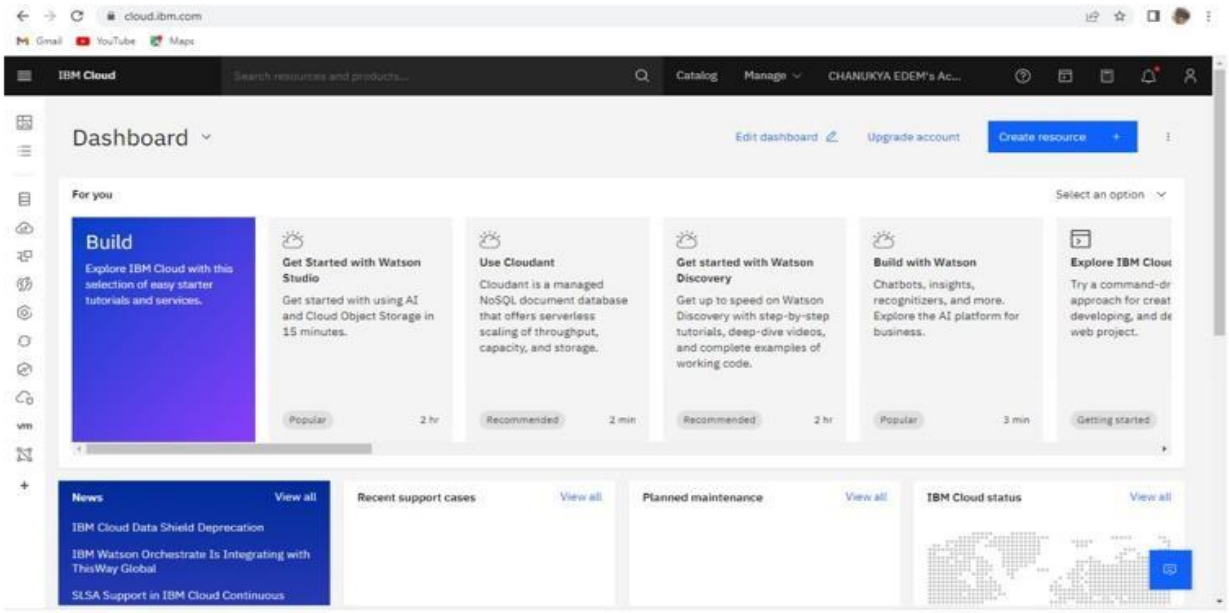
import requests from
tensorflow.keras.preprocessing import image from
tensorflow.keras.models import load_model
import numpy as np import pandas as pd import
tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for import
os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)

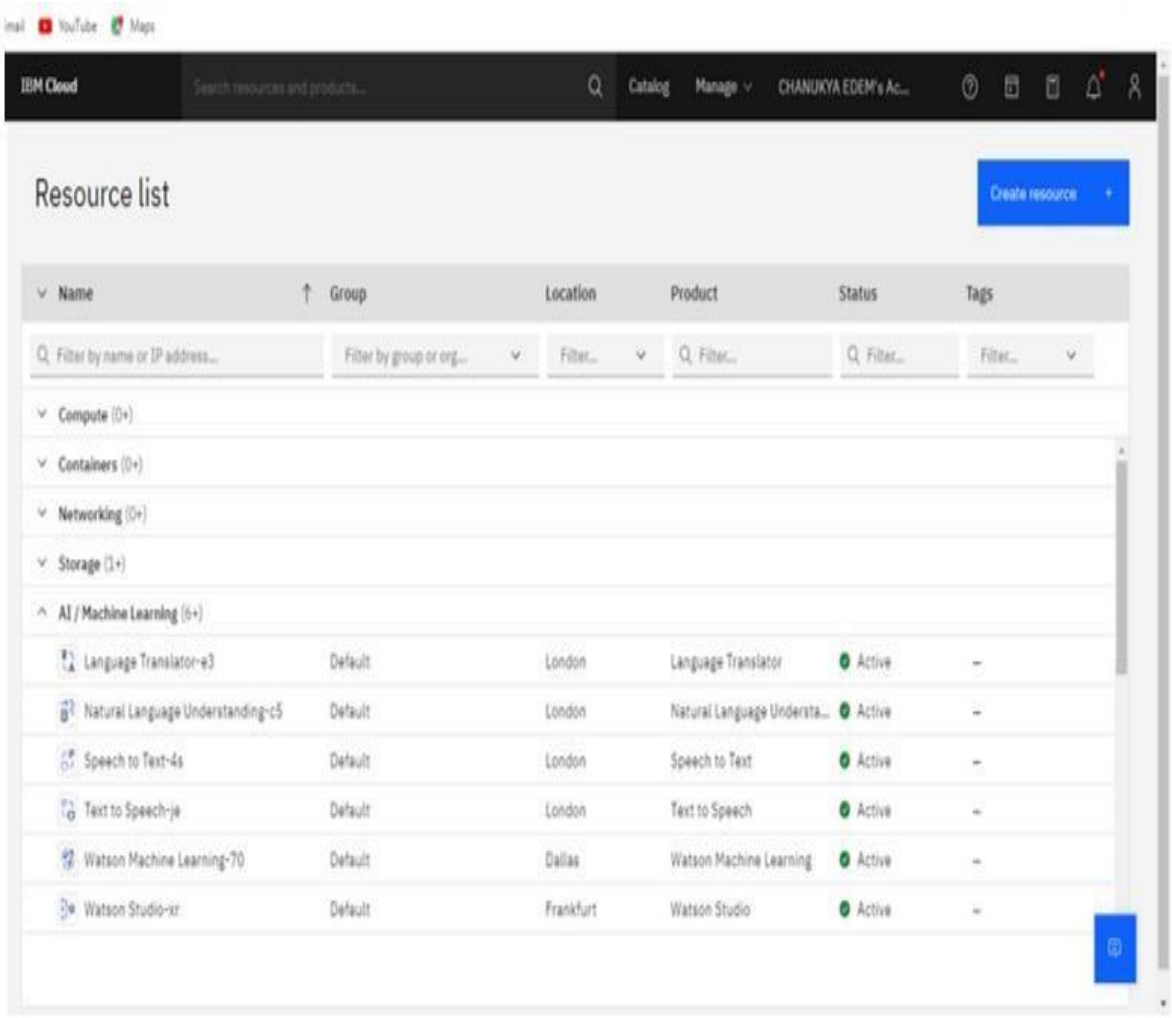
```
#load both the vegetable and fruit models model
= load_model("vegetable.h5")
model1=load_model("fruit.h5")

#home page @app.route('/') def
home():     return
render_template('home.html')

#prediction page @app.route('/index')
def prediction():     return
render_template('predict.html')

@app.route('/index',methods=['POST'])
          def predict():     if
request.method == 'POST':         # Get
the file from post request
       f = request.files['image']

       # Save the file to ./uploads
basepath = os.path.dirname(__file__)
       file_path = os.path.join(
          basepath, 'uploads', 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=="vegitable"):
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)
```

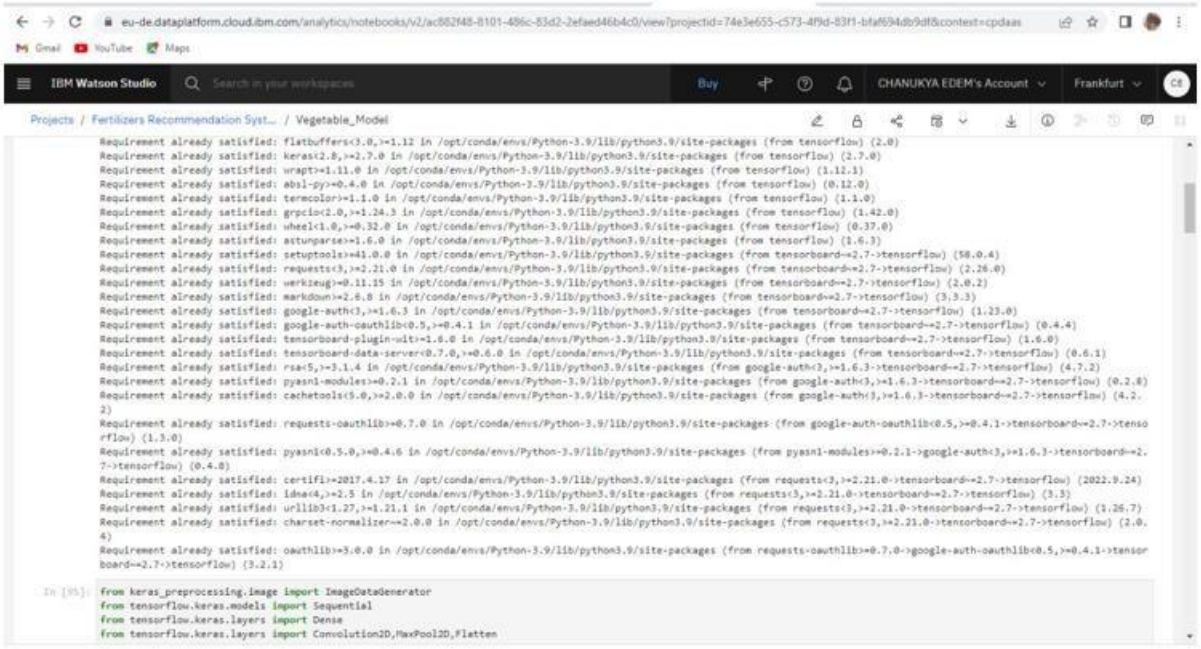
```
df=pd.read_excel('precautions - fruits.xlsx')
print(df.iloc[preds]['caution'])


        return df.iloc[preds]['caution'] if
__name__ == "__main__":
    app.run(debug=False)
```

ADDING TO IBM CLOUD DEPLOYMENT:



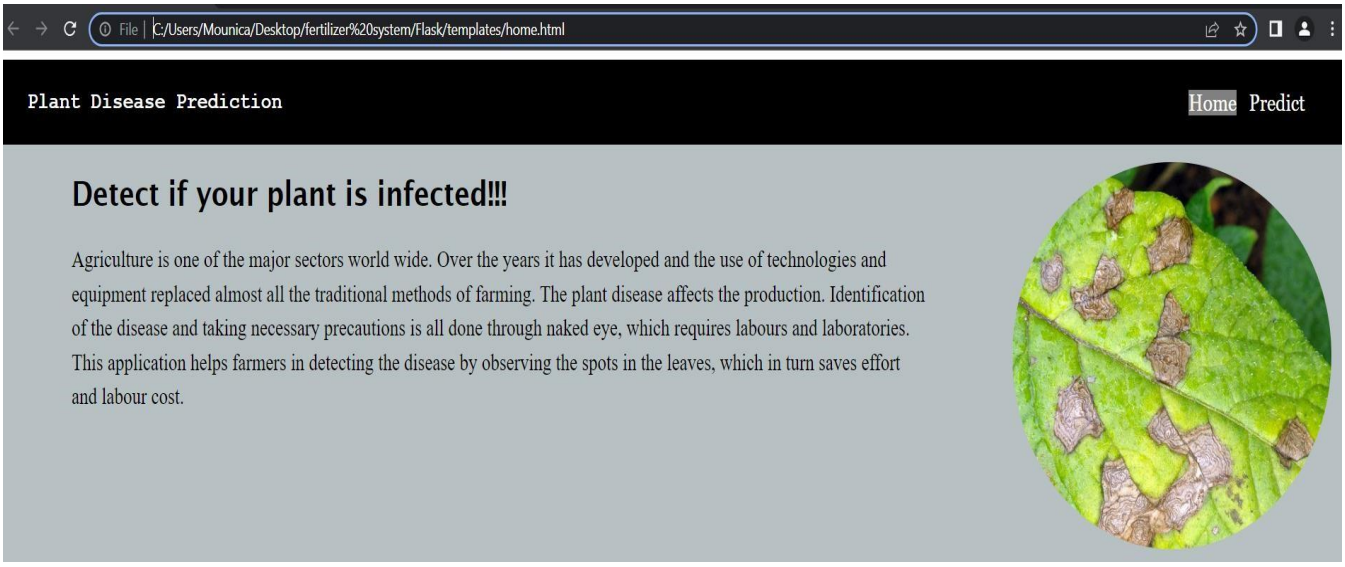
IBM CLOUD RESOURSE LIST:

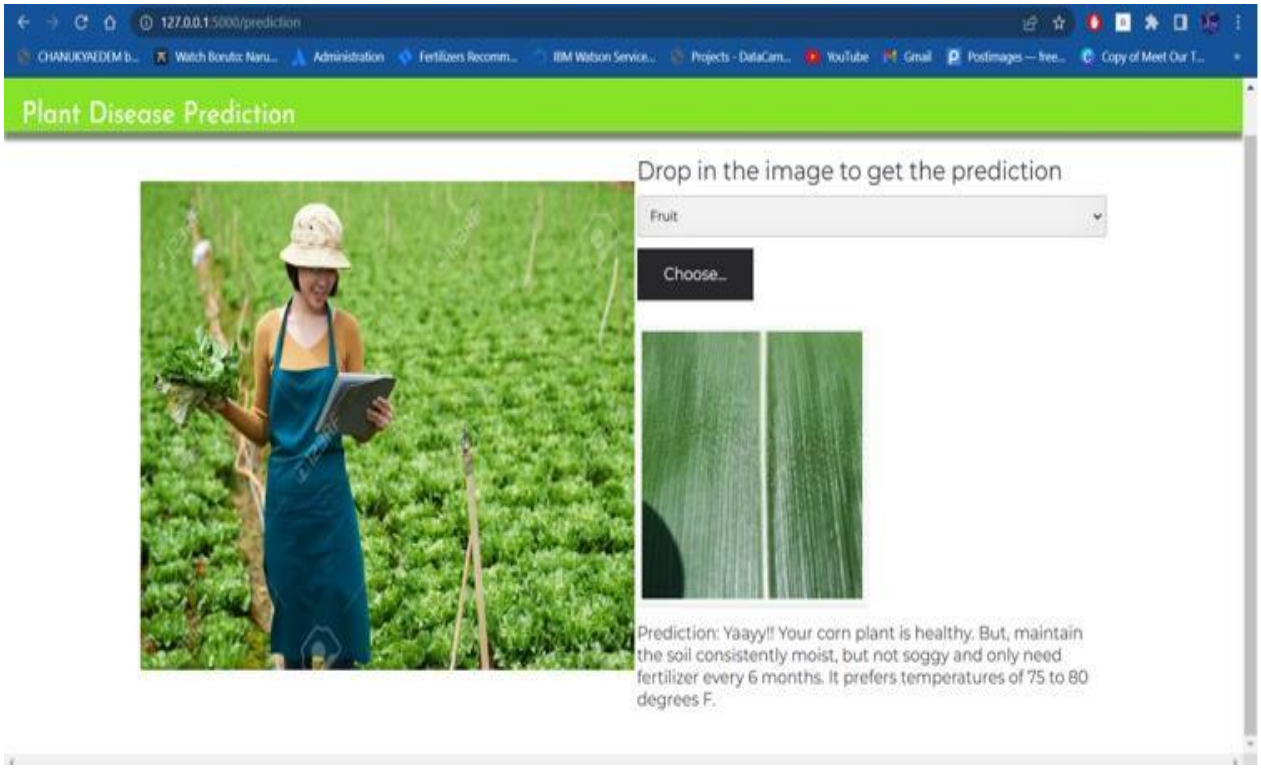

YouTube
Maps
IBM Cloud
Search resources and products...
Catalog
Manage
CHANUKYA EDEM's Ac...
Resource list
Create resource
Name
Group
Location
Product
Status
Tags
Filter by name or IP address...
Filter by group or org...
Filter...
Compute (0+)
Containers (0+)
Networking (0+)
Storage (1+)
AI / Machine Learning (6+)
Language Translator-e3
Default
London
Language Translator
Active
Natural Language Understanding-c5
Default
London
Natural Language Understa...
Active
Speech to Text-4s
Default
London
Speech to Text
Active
Text to Speech-je
Default
London
Text to Speech
Active
Watson Machine Learning-70
Default
Dallas
Watson Machine Learning
Active
Watson Studio-xr
Default
Frankfurt
Watson Studio
Active

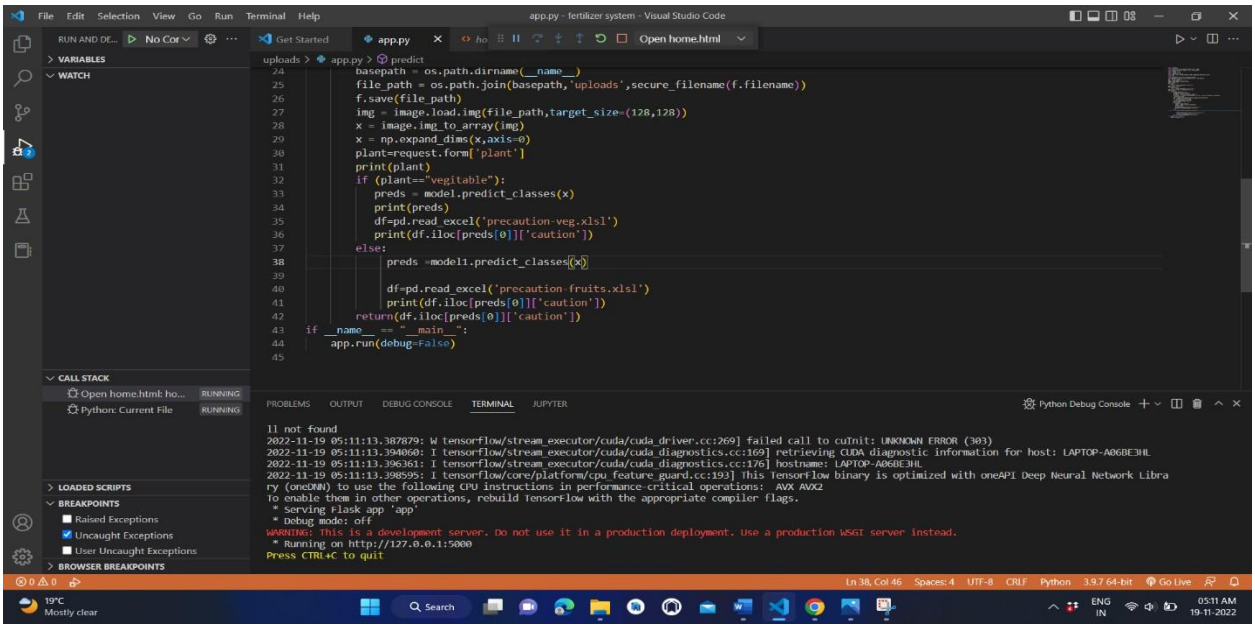

eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/ac882f48-8101-486c-83d2-2efaed46b4c0/view?projectid=74e3e655-c573-4f9d-83f1-bfaf694db9df&context=cpdaas
Gmail
YouTube
Maps
IBM Watson Studio
Search in your workspaces
Buy
CHANUKYA EDEM's Account
Frankfurt
Projects / Fertilizers Recommendation Syst... / Vegetable_Model
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.0)
Requirement already satisfied: keras<2.8,>=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: absl-py>=0.4.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.12.0)
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: wheel<1.0,>=0.32.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.37.0)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: setuptools>=41.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (58.0.4)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (2.26.0)
Requirement already satisfied: werkzeug>=0.11.15 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (2.0.2)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (3.3.3)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (1.23.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (0.4.4)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (1.6.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (0.6.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.7->tensorflow) (4.7.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.7->tensorflow) (0.2.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.7->tensorflow) (4.2.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.7->tensorflow) (1.3.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.7->tensorflow) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (2.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.7->tensorflow) (3.2.1)
In [95]:
from keras_preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D,MaxPool2D,Flatten

7.2 Features 2:

HTML PAGES FOR WEB APPLICATION:






File Edit Selection View Go Run Terminal Help
app.py - fertilizer system - Visual Studio Code
app.py
Open home.html
uploads > app.py > predict
basepath = os.path.dirname(__name__)
file_path = os.path.join(basepath,'uploads',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=="vegitable"):
preds = model.predict_classes(x)
print(preds)
df=pd.read_excel('precaution-veg.xlsl')
print(df.iloc[preds[0]]['caution'])
else:
preds =model1.predict_classes(x)
df=pd.read_excel('precaution-fruits.xlsl')
print(df.iloc[preds[0]]['caution'])
return(df.iloc[preds[0]]['caution'])
if __name__ == "__main__":
app.run(debug=False)
VARIABLES
WATCH
CALL STACK
Open home.html: ho... RUNNING
Python: Current File RUNNING
LOADED SCRIPTS
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
BROWSER BREAKPOINTS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python Debug Console
ll not found
2022-11-19 05:11:13.387879: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-19 05:11:13.394060: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-A06BE3HL
2022-11-19 05:11:13.396361: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-A06BE3HL
2022-11-19 05:11:13.398595: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Libra
ry (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
Ln 38, Col 46 Spaces: 4 UTF-8 CRLF Python 3.9.7 64-bit Go Live
19°C Mostly clear
Search
ENG IN
05:11 AM 19-11-2022

8.Testing

8.1 Test Case:

H9 | When user clicks the classify option, the classified result should be displayed.

				Date	03-Nov-22									
				Team ID	PNT2022TMID43438									
				Project Name	Fertilizer recommendation System for disease prediction									
				Maximum Marks	4 marks									
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automatio	BUG ID	Executed By	
									Pass					
HOME PAGE	Functional	Home page	verify user is able to see the predict popup when user clicked on button		1.Enter URL and click go 2.Click on button in home page 3.verify predict page popup displayed or not	https://127.0.0.1:5000	Predict popup should display	Working as expected	Pass					
PREDICT PAGE	Functional	Predict page	train ml model and labels in html to get	Create an upload file option	1. 2. Click on the Upload an image option. 3. Upload an image to be classified. 4. Click on classigy option to classify the image.	https://127.0.0.1:5000	When user clicks the classify option, the classified result should be displayed.	Working as expected	Pass					

User Acceptance:

1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] 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	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6

Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3.Test Case Analysis

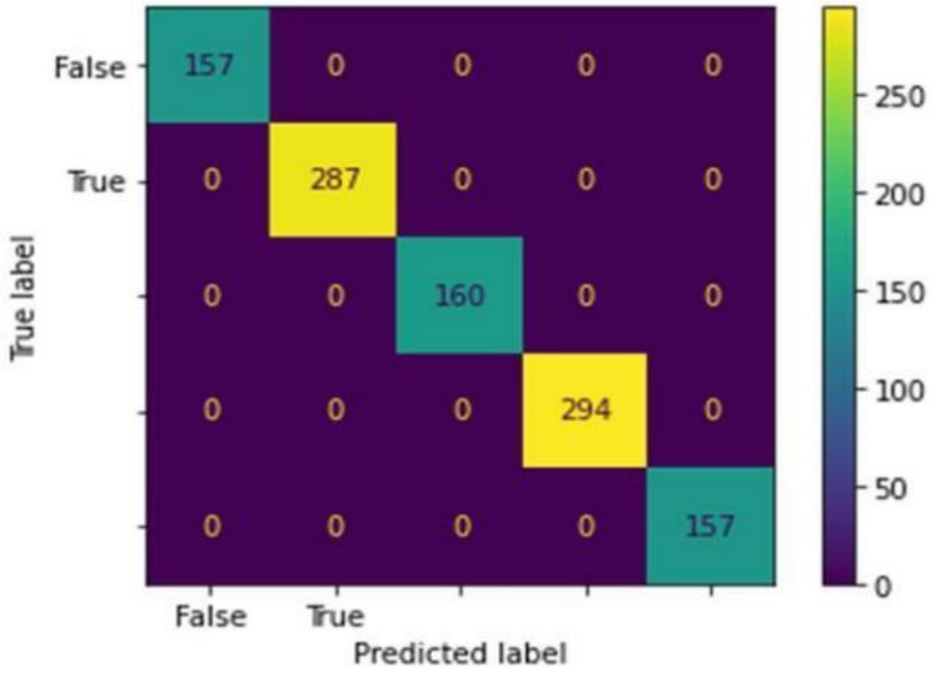
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. Result

9.1 Performance Metrices:

1. Confusion Matrix



```
print(metrics.classification_report(test_data['label'].values, test_data['model_preds'].values
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       157
           1       1.00      1.00      1.00       287
           2       1.00      1.00      1.00       160
           3       1.00      1.00      1.00       294
           4       1.00      1.00      1.00       157

    accuracy                           1.00      1055
   macro avg       1.00      1.00      1.00      1055
weighted avg       1.00      1.00      1.00      1055
```

2. Accuracy – 100 %

```
[8] print(f"the accuracy is {metrics.accuracy_score(test_data['label'].values, test_data['model_preds'].values)}")
```

```
the accuracy is 1.0
```

3. Precision – 100 %

```
[11] print(f"the precision is {metrics.precision_score(test_data['label'].values, test_data['model_preds'].values, average = 'weighted')}")
```

```
the precision is 1.0
```

4. Recall – 100 %

```
[12] print(f"the recall is {metrics.recall_score(test_data['label'].values, test_data['model_preds'].values, average = 'weighted')}")
```

```
the recall is 1.0
```

5. Specificity – 100 %

```
print(f"the specificity is {metrics.recall_score(test_data['label'].values, test_data['model_preds'].values, pos_label=0,average = 'weighted')}")
```

```
the specificity is 1.0
```

6. F1-Score – 100 %

```
[13] print(f"the f1 score is {metrics.f1_score(test_data['label'].values, test_data['model_preds'].values,average = 'weighted')}")
```

```
the f1 score is 1.0
```

10.Advantages and Disadvantages:

10.1 Advantages:

The system comes with a model to be precise and accurate in predicting crop yield and deliver the end user with proper recommendations about required fertilizer ratio based on atmospheric and soil parameters of the land which enhance to increase the crop yield and increase farmer revenue. The prediction of crop yield based on location and proper implementation of algorithms have proved that the higher crop yield can be achieved. From above work I conclude that for soil classification Random Forest is good with accuracy 86.35% compare to Support Vector Machine. For crop yield prediction Support Vector Machine is good with accuracy 99.47% compare to Random Forest algorithm. The work can be extended further to add following functionality. Mobile application can be build to help farmers by uploading image of farms. Crop diseases detection using image processing in which user get pesticides based on disease images. Implement Smart Irrigation System for farms to get higher yield.

- Fertilizers have all nutrients required for plants growth.
- It is soluble and easily absorbed by plants.
- It enhances the metabolism of plants.
- It is easily available in the market.
- Highly needed for large production.

10.2 DISADVANTAGE:

- Fertilizers are more expensive than manure.
- Over fertilization can damage the plants.
- It is toxic and can harm humans.
- It affected the environment and echo system.
- Long term use reduce soil quality.

11.Conclusion

The core strategy of this project is to predict the crop based on the soil nutrient content and the location where the crop is growing. This system will help he farmers to choose the right crop for their land and to give the suitable amount of fertilizer to produce the maximum yield. The Support Vector Machine algorithm helps to predict the crop the precisely based on the pre-processed crop data. This system will also

help the new comers to choose the crop which will grow in their area and produce them a good profit.

A decent amount of profit will attract more people towards the agriculture. Also, the crop growth is based on the climate conditions in the particular area and the seasonal monsoons happens now are unpredictable, hence it is easy for the farmers when the prediction result is also based on the climatic conditions. Live weather prediction will also help the users to predict the crop water needs and also it will help the farmers to decrease the crop damage due to the rain or drought

The prediction of crop yield based on soil data and proper implementation of algorithms have proved that a higher crop yield can be achieved. From the above work, we conclude that for soil classification Random Forest is a suitable algorithm with an accuracy of 99.09% compare to Gaussian Naive Bayes. The work can be extended further to add the following functionality. Building a Website can be built to help farmers by uploading an image of farms. Crop diseases detection uses image processing in which users get pesticides based on disease images and Fertilizer prediction based on soil condition.

By categorizing the soil samples according to the soil type, land type and macro nutrients Nitrogen (N), Phosphorus (P) and Potassium (K) present in the soil the suitable crop along with its appropriate fertilizer is suggested to the agricultural stakeholder. The month in which the yield will be high is also suggested to the user. The yield calculation is also provided for the crop selected by the farmer. The proposed crop recommendation system provides 82% of accuracy.

12.Future Scope

The future work is to implement Machine Learning Algorithms like Ensemble Classifiers to predict the crop yield and recommend the crop with appropriate fertilizer. In the existing system only soil characteristics were considered to provide crop recommendations. In the future work the climatic parameters will also be taken into account to provide crop recommendations. Also the method can be extended to include diverse varieties of crop to be cultivated and to analyse it's performance.

This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy. The proposed algorithm can be modified further to identify the disease that affects the various plant organs such as stems and fruits.

13.Appendix

13.1 Source code:

PYTHON CODE FOR FLASK:

```
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__)

#load both the vegetable and fruit models model
= load_model("vegetable.h5")
model1=load_model("fruit.h5")

#home page
@app.route('/') def
home():
   return render_template('home.html')

#prediction       page
@app.route('/index')
def prediction():
   return render_template('predict.html')

@app.route('/index',methods=['POST'])
          def predict():     if
request.method == 'POST':         # Get
the file from post request
       f = request.files['image']
```

```
        # Save the file to ./uploads
basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', 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=="vegitable"):
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'] if
__name__ == "__main__":
app.run(debug=False)
```

HTML CODE:

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <link rel="stylesheet" href="../static/style1.css">
</head>

<body>
   <header class="main-content">
      <div class="plant-disease">
         <div class="plant-topic">
            <span class="plant">Plant Disease Prediction</span>
</div>
         <div class="topic-right">
            <a class="menu" href="">Home</a>
            <a class="menu-2" href="../templates/index.html">Predict</a>
</div>
      </div>
   </header>
   <section class="Disease-content-1">
      <div class="content">
         <div class="content-topic">
            <h1>Detect if your plant is infected!!!</h1>
            <p class="Details">Agriculture is one of the major sectors world wide. Over the years it has developed
               and the use of
               technologies and equipment replaced almost all the traditional methods of farming. The plant disease
               affects the production. Identification of the disease and taking necessary precautions is all done
               through naked eye, which requires labours and laboratories. This application helps farmers in
               detecting the disease by observing the spots in the leaves, which in turn saves effort and labour
               cost.</p>
         </div>
         <div class="img">
            <img class="img-1" src="../templates/potato-blight (1).jpg" alt="">
</div>
```

```
      </div>
   </section>


</body>
```

Index.html:

```
</html>
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <link rel="stylesheet" href="../static/style.css">
</head>
<body>
   <section class="Disease-content-2">
      <div class="Disease-topic">
         <div class="Details-menu">
            <h4>Plant Disease Prediction</h4>

         </div>
         <div class="content-menu">
            <div class="disease-img">
               <img     class="img-2"     src="../templates/img     right.jpg">
</div>
            <div class="content-2">
               <div class="img-left">
                  <span class="Drop">Drop in the image to get the prediction</span>
                  <button class="tag"> Fruit</button>
            <form>
               <label for="myfile"></label>
               <input type="file" id="myfile" name="myfile">
               </div>
```

```
                <div class="img-right">

                    <img class="img-3" src="../templates/potato-blight (1).jpg"alt="">
<button class="predict">predict</button>                    </div>
            </div>
        </div>


    </div>
  </section>
</body>

</html>
```

13.2 Github and Project Demo Link:

Github link:

https://github.com/IBM-EPBL/IBM-Project-42211-1660656292

Demo Link:
https://drive.google.com/file/d/1cZ9y7DUxqIQ8NMjNmLHHc3ASahYqQrJG/view?usp=share_link