Project Report Format

Team Id	**PNT2022TMID19575**
Project Name	**Fertilizer Recommendation System for Disease Prediction**

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

Project Report

1. INTRODUCTION

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.

1.2 Purpose:

Crop fields being affected by various types of disease leads to health issues for the consumers and also result in less production of crops, which leads to financial crises. In order to overcome this we need to find the disease at an early stage and recommend a suitable fertilizer for the crops. So our project helps the farmers to identify the diseases of the affected plant just by uploading the image of it.

2. LITERATURE SURVEY:

2.1 Existing problem:

Agriculture plays an essential part in an economy's life. They are the backbone of our country's economic system. One of the key problems confronting farmers is selecting the right crop for cultivation. The selection of crops is determined by several factors such as temperature, soil composition, market prices, etc. Machine Learning is a technique that uses complex algorithms and a collection of predefined rules to operate intelligently. It uses past data to read the patterns and then performs the intended task according to the defined rules and algorithms based on the analysis it produces. In this project, a system is developed in which a Voting Based Ensemble Classifier is applied to recommend the appropriate crops. This system also proposes the required fertilizer to boost the nutrients contained in the soil and thus enhance the yield of the crop. Thus, there arises a need for suggesting suitable crops and fertilizers using a machine learning algorithm.

2.2 Reference:

[1] Mansi Shinde, Kimaya Ekbote, Sonali Ghorpade, "Crop Recommendation and Fertilizer Purchase System", IJCSIT International Journal of Computer Science and Information Technologies, Volume 7, Issue 2, 2016.

[2] V. Sellam, E. Poovammal, "Prediction of Crop Yield using Regression Analysis", IJST Indian Journal of Science and Technology, Volume 9, Issue 38, October 2016.

[3] U.K. Diwan, H.V. Puranik, G.K. Das, J.L. Chaudhary, "Yield Prediction of Wheat at Pre-Harvest Stage Using Regression-Based Statistical Model for 8 District of Chhattisgarh, India",

IJCMAS International Journal of Current Microbiology and Applied Sciences, Volume 7, Issue 1,2018.

[4] Rushika Ghadge, Juilee Kulkarni, Pooja More, Sachee Nene, Priya.RL, “Prediction of Crop Yield using Machine Learning”, IRJET International Research Journal of Engineering and Technology, Volume 5, Issue 2, February 2018.

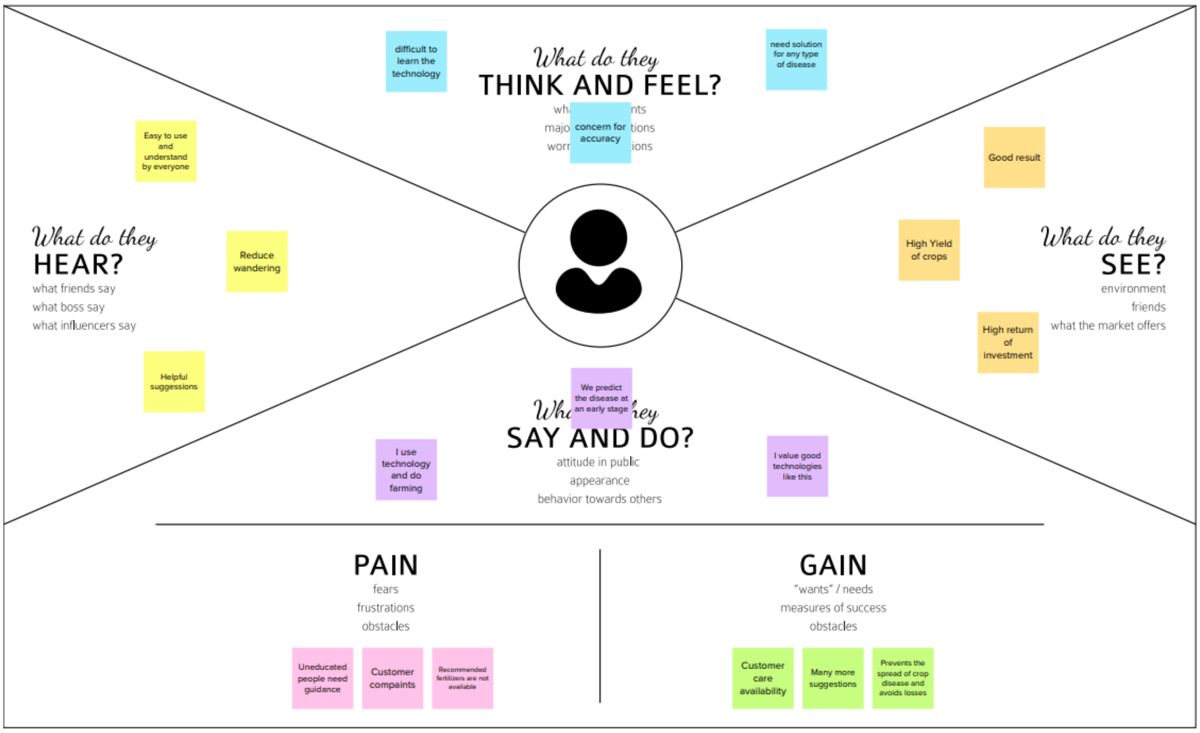
[5] P. Priya, U. Muthaiah, M. Balamurugan, “Predicting yield of the crop using machine learning algorithm”, IJESRT International Journal of Engineering Sciences & Research Technology, Volume 7, Issue 4, April 2018.

2.3 Problem Statement Definition:

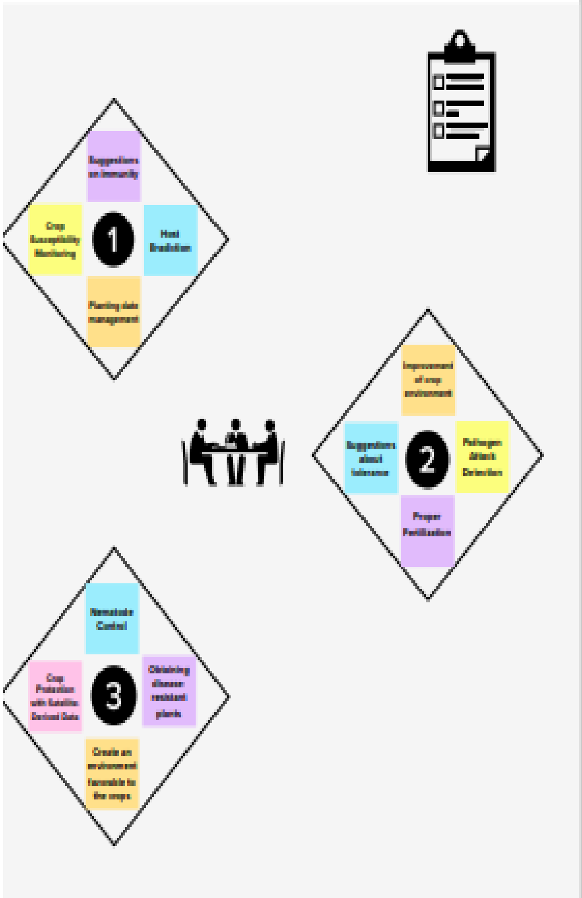
This project is about identifying various types of disease in crop fields (i.e., Fruits and Vegetable plants) and recommending suitable fertilizers by processing the various images using AI technology.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



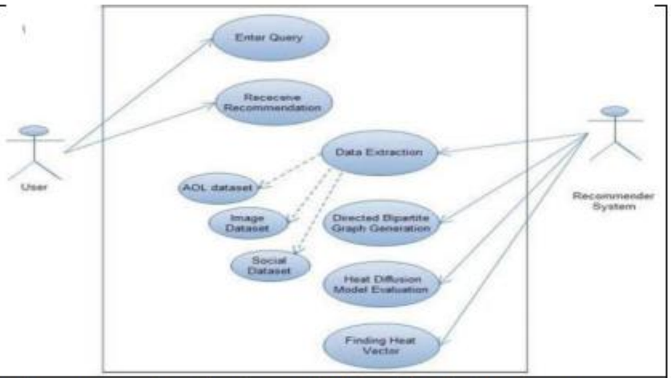
3.2 Ideation & Brainstorming:


Suggestions on immunity
Crop Susceptibility Monitoring
1
Host Eradiction
Planting date management
Improvement of crop environment
Suggestions about tolerance
2
Pathogen Attack Detection
Proper Fertilization
Nematode Control
Crop Protection with Satellite Derived Data
3
Obtaining disease resistant plants
Create an environment favorable to the crops


Pathogen Attack Detection
Suggestions on immunity
Nematode Control
Host Eradiction
Improvement of crop environment
NDWI maps for monitoring plant water content
Proper Fertilization
Crop Protection with Satellite Derived Data
Disease Control with Satellite Derived Data
Suggestions about tolerance
Abiotic Stress Level Indicators
Proper Fertilization
Discussions on crop resistance
Importance
Feasibility

3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves. Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest.
2.	Idea / Solution description	Leaves are affected by bacteria, fungi, virus, and other insects. Support Vector Machine (SVM) algorithm classifies the leaf image as normal or affected. Vectors are constructed based on leaf features such as colour, shape, textures. Then hyperplane constructed with conditions to categorize the pre-processed leaves and also implement multiclass classifier, to predict diseases in leaf image with improved accuracy.
3.	Novelty / Uniqueness	Recommends the fertilizer for affected leaves based on severity level. Fertilizers may be organic or inorganic. Admin can store the fertilizers based on disease categorization with severity levels. The measurements of fertilizers are also suggested based on disease severity.
4.	Social Impact / Customer Satisfaction	Presently our farmers are not effectively using technology and analysis, so there may be a chance of wrong selection of fertilizer for crops that will reduce their income. To reduce those type of loses we have developed a farmer friendly system with GUI, that will predict which would be the best suitable fertilizer for particular crop disease. So, this makes the farmers to take right decision in selecting the fertilizer for crop disease such that agricultural sector will be developed by innovative idea.

5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	The proposed method uses SVM to classify tree leaves, identify the disease and suggest the 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 existing CNN. For the same set of images, F-Measure for CNN is 0.7and 0.8 for SVM, the accuracy of identification of leaf disease of CNN is 0.6 and SVM is 0.8.

3.4 Problem Solution fit:

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Farmers are the customers of our project who get benefits by using our prediction system which predicts the perfect fertilizer for plant diseases that can be used on affected plant to cure diseases.

6. CUSTOMER CONSTRAINTS CC

Some people may find it difficult to understand the application for the first time of use

5. AVAILABLE SOLUTIONS AS

As we predict the disease at an early stage and recommend a fertilizer with the location details such as where it is available, this project will become perfect solution for customers.

Explore AS, differentiate

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

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

Generally, it's estimated that various pests (insects, weeds, nematodes, animals, diseases) each yearcause crop yield losses of 20-40%.In order to avoid this, earlier prediction is necessary. Although our 1st preference will be given to the major food yielding crops such as wheat , rice followed by others.

9. PROBLEM ROOT CAUSE RC

Infectious plant diseases are mainly caused bypathogenic organisms such as fungi, bacteria, viruses, protozoa, as well as insects and parasitic plants

7. BEHAVIOUR BE

First we have to provide a clear overview of how our application is going to work i.e., just by uploading an image of the crops, the disease prediction is done and the customers can easily get the fertilizer recommendation for the affected crops.

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

Identify strong TR & EM

3. TRIGGERS TR

We can show our customers about the ratings and reviews of other customers and this will lead to the high usage of our services

4. EMOTIONS: BEFORE / AFTER EM

There will be mixed responses at the beginning stages of our application, some may find it easy and convenient to use others may find it difficult to use or may be they even find it difficult how it works .we can overcome this by teaching them how the application works

10. YOUR SOLUTION SL

1. By explaining the customers about how to use this services by instructions help them to learn the application easily.
2. Displaying the impact of the particular disease during the disease prediction will reduce the unawareness of that problem.
3. Enabling the ratings and review options.
4. Adding the customer support page to contact the customer care in case of any problems.
5. Increase accuracy using Machine Learning technique.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

The customers could learn to use the application, so that they can get an efficient results.

8.2 OFFLINE

They should aware of the seriousness and follow the recommendations properly

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	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login with user name Login with password
FR-4	Profile update	Update the user credentials Update the Contact details
FR-5	Uploading Images	Capture the image of the affected Crop Upload the image of the affected Crop This model will predict the disease of the affected Crop
FR-6	Recommendation	User will request the fertilizer Get the fertilizer recommendations
FR-7	Ratings and Reviews	Share their Experiences Give the Feedback

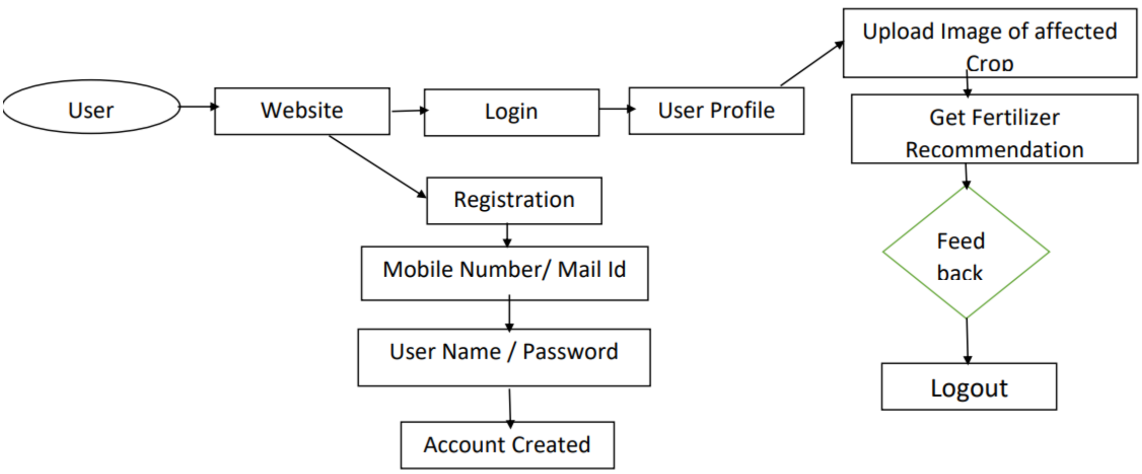
4.2 Non-Functional requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This service is designed and can be used on both website and mobile browsers so that the usability of this application is very efficient.
NFR-2	**Security**	This can be used only by users who have their proper login credentials
NFR-3	**Reliability**	In case of any issues such as the delay in the responses, it will be rectified to maintain its reliability.
NFR-4	**Performance**	Sometimes the wrong predictions occur due to the inaccuracy of the model at a rare point, in order to rectify this, this application will run the model more than one time to predict the exact result and recommends the fertilizer for that disease.
NFR-5	**Availability**	It will predict any type of new disease by learning from the available dataset and predict the disease accurately.
NFR-6	**Scalability**	It can be accessed by more number of users at the same time without any performance issues.

5. PROJECT DESIGN:

5.1Data Flow Diagrams:



5.2 Solution & Technical Architecture:

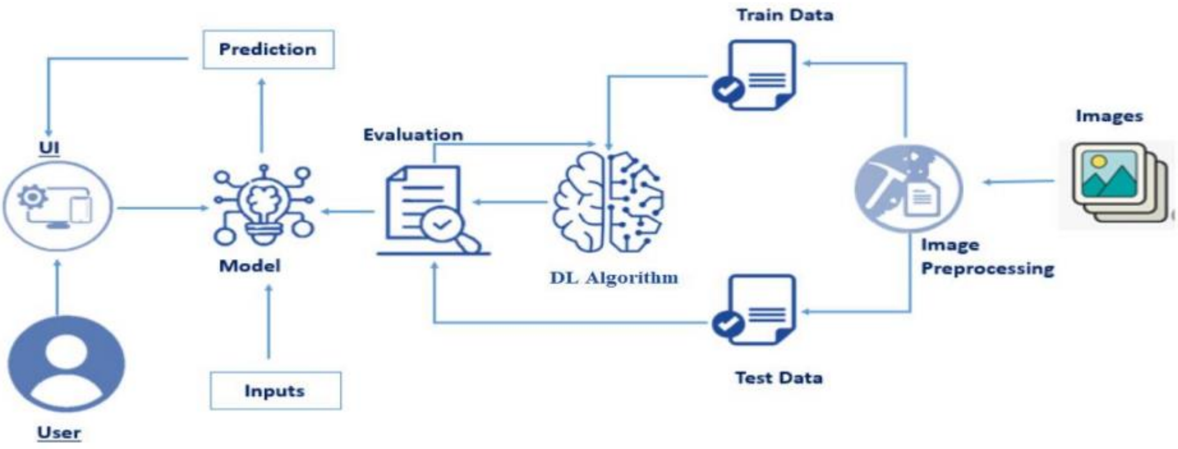


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI.	HTML, CSS, JavaScript
2.	Application Logic-1	The user can interact with the web UI through the general instructions for the first time of their usage.	Python
3.	Application Logic-2	The customers can login into their applications and can upload their images of the crops and ask for the suggestions of fertilizers on the display.	IBM Watson STT service
4.	Application Logic-3	This model will detect the disease of the affected crops and convey the information to the users through the assistants.	IBM Watson Assistant
5.	Database	Image data sets.	MySQL.
6.	Cloud Database	Database Service on Cloud	IBM Cloudant etc.
7.	File Storage	Image files storage system.	Local Filesystem
8.	External API-1	A software is used to process the crop images and detect the diseases.	Anaconda software.
9.	External API-2	The external API is used to recommend the suitable fertilizers for the affected crops.	Jupyter notebooks.
10.	Machine Learning Model	The part of the machine learning model called the Deep Learning model is used here to process the various images and the user uploaded images to identify the absolute disease at the early stage to avoid the yield losses.	Image Recognition Model.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Django Flask	Python web framework.
2.	Security Implementations	Use of firewalls 2 step verifications.	Encryption algorithms.
3.	Scalable Architecture	Better user experience and higher agility.	SAAS services like ECS, data lakes.
4.	Availability	Use of load balancers and elastic storage systems.	Elastic storage
5.	Performance	The performance of the application is enhanced by the various techniques that are widely used for processing the data such as the number of requests per sec and use of Cache is efficiently managed.	Distributed architecture.

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	Data collection	USN-1	As a user, I can collect the datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc. with different vegetable leaf images.	5	Low	PAVITHRA M GNANAPRAVEENA R
Sprint-1		USN-2	As a user, I can collect the dataset from different open sources with different fruit leaf images.	5	Low	SNEKA K VAISHNAVI K
Sprint-1	Image Pre-processing	USN-3	As a user, I have to pre-process the images and then feed them onto the model for training	10	Medium	VAISHNAVI K GNANAPRAVEENA R PAVITHRA M
Sprint-2	Model building for fruit disease prediction.	USN-4	As a user, I will get an application with the DL model that will provide information about fruit diseases.	4	Medium	SNEKA K PAVITHRA M
Sprint-2	Model building for vegetable disease prediction.	USN-5	As a user, I will get an application with the DL model that will provide information about vegetable diseases.	4	High	GNANAPRAVEENA R VAISHNAVI K
Sprint-2	Add CNN layer	USN-6	As a user, I will be adding three layers for CNN which include the Convolution layer, Pooling layer, and Flattening layer	4	Medium	VAISHNAVI K
Sprint-2	Add dense layer	USN-7	Creating the model and adding the input, hidden, and dense layer to it.	4	Low	GNANAPRAVEENA R
Sprint-2	Train and save the model	USN-8	As a user, I can compile, fit and save the model.	2	Medium	SNEKA K
Sprint-2	Test both the models	USN-9	The model is to be tested with different images to know if it is working correctly by loading	2	High	PAVITHRA M

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			the test image, pre-processing it, and predicting the diseases.			
Sprint-3	Application building	USN-10	As a user, I will upload the image of the affected leaf by clicking the upload button in the user interface.	10	High	VAISHNAVI K SNEKA K
Sprint-3		USN-11	As a user, I will request the application to predict the disease that affected my plant.	1	Medium	GNANAPRAVEENA R VAISHNAVI K
Sprint-3		USN-12	As a user, I will get information about the particular disease and also suggestions for fertilizers for that disease.	9	Low	PAVITHRA M VAISHNAVI K
Sprint-4	Train the model on IBM	USN-13	As a user, I train the model on IBM and built the deep learning model.	10	High	GNANAPRAVEENA R PAVITHRA M SNEKA K VAISHNAVI K
Sprint-4	Cloud deployment	USN-14	As a user, I will deploy the application onto the cloud which is to be accessed by users from everywhere.	10	High	GNANAPRAVEENA R PAVITHRA M SNEKA K VAISHNAVI K

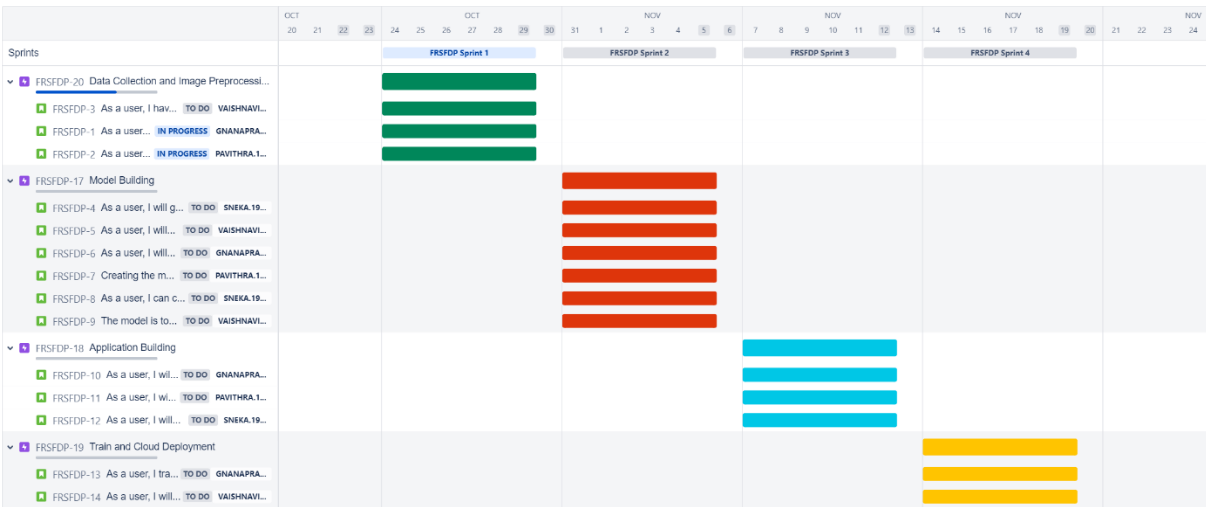
6.2 Sprint Delivery Schedule:

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

Velocity:

$$Average\ velocity = \frac{Sprint\ duration}{Velocity} = \frac{20}{6} = 3.33$$

6.3 Reports from JIRA:



7. CODING AND SOLUTIONING:

FRUITS:

```
import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-
out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the
regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #convolutional Layer
```

```python
from keras.utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt   #used for data visualization
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='cfsqIQ3AGixnhhrmseDPNd6p657ZdFzOaF0noHqsK2xG',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'diseaseprediction-donotdelete-pr-xuizddnjq94zqq'
object_key = 'fruit-dataset.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
pwd
import os
filenames=os.listdir('/home/wsuser/work/fruit-dataset/train')
x_train = train_datagen.flow_from_directory('/home/wsuser/work/fruit-
dataset/train',target_size=(64,64),class_mode='categorical',batch_size=100)
x_test = train_datagen.flow_from_directory('/home/wsuser/work/fruit-
dataset/test',target_size=(64,64),class_mode='categorical',batch_size=100)
x_train.class_indices
model = Sequential()
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300,activation='relu')) # Hidden layer
model.add(Dense(150,activation='relu')) # Hidden layer
model.add(Dense(units = 6,activation='softmax'))
```

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accur
acy'])
len(x_train)
model.fit(x_train,
          steps_per_epoch=len(x_train),
          epochs=10,
          )
model.save("fruits_train.h5")
!tar -zcvf fruits-model_new.tgz fruits_train.h5
ls -1
!pip install watson-machine-learning-client --upgrade
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials ={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"9Mu65MQBe6ruThRUvbGPUQ_87a9ZFJU5oMrWNCogAk6N"
}
client = APIClient(wml_credentials)
client.spaces.get_details()
def guid_from_space_name(client,deploy):
  space = client.spaces.get_details()
  return (next(item for item in space['resources'] if
item['entity']['name']==deploy)['metadata']['id'])
space_uid = guid_from_space_name(client,'Fruits_classification')
print("Space UID = " + space_uid)
client.set.default_space(space_uid)
client.software_specifications.list(limit=100)
software_space_uid =
client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
model_details = client.repository.store_model(model='fruits-
model_new.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"CNN",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
model_details
model_id = client.repository.get_model_id(model_details)
model_id
client.repository.download(model_id,'fruits_model_new.tar.gz')
```

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
from PIL import Image
import numpy as np
model = load_model("fruits_train.h5")
```

8.**TESTING FOR FRUITS PREDICTION:**

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='cfsqIQ3AGixnhhrmseDPNd6p657ZdFzOaF0noHqsK2xG',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'diseaseprediction-donotdelete-pr-xuizddnjq94zqq'
object_key = '0da48999-25da-4373-8277-3718a8203d0a___RS_HL 8012.JPG'

streaming_body_4 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']
```

```
img = Image.open(streaming_body_4).convert("L")
img = img.resize( (64,64) )
img
x = np.array(img)
x=np.ones((64,64,3))
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=["Apple___Black_rot","Apple___healthy","Corn_(maize)___Northern_Leaf_Bli
ght","corn blight","Peach___Bacterial_spot", "Peach___healthy"]
preds=(index[y[0]-2])
preds
```

VEGETABLES CODING:

```
   pwd
!pip install tensorflow --upgrade
import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-
out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the
regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #convolutional Layer
from keras.utils import np_utils #used for one-hot encoding
```

```
import matplotlib.pyplot as plt   #used for data visualization
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='o58uAinmdzXtLgQHZVw8WeFds-HPV_FiqQno1ZAFECSp',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'vegetablediseaseprediction-donotdelete-pr-tu6rk1bz8j3qu4'
object_key = 'Veg-dataset.zip'

streaming_body_5 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']

from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_5.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
pwd
import os                                                              b
filenames=os.listdir('/home/wsuser/work/Veg-dataset/train_set')
x_train = train_datagen.flow_from_directory('/home/wsuser/work/Veg-
dataset/train_set',target_size=(64,64),class_mode='categorical',batch_size=100
)
x_test = train_datagen.flow_from_directory('/home/wsuser/work/Veg-
dataset/test_set',target_size=(64,64),class_mode='categorical',batch_size=100)
x_train.class_indices
model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
model.add(Dense(300,activation='relu')) # Hidden layer
model.add(Dense(150,activation='relu')) # Hidden layer
model.add(Dense(units = 9,activation='softmax'))
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accur
acy']
len(x_train)
model.fit(x_train,
          steps_per_epoch=len(x_train),
          epochs=10,
          )
model.save("vegetables_train.h5")
!tar -cf vegetables-model_new.tgz vegetables_train.h5
ls -1
!pip install watson-machine-learning-client --upgrade
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials ={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"Jl0VgQ40vRcibWK6COsvcy_bqgbFXU3noYZRBiEwgGEb"
}
client = APIClient(wml_credentials)
def guid_from_space_name(client,deploy):
  space = client.spaces.get_details()
  return (next(item for item in space['resources'] if
item['entity']['name']==deploy)['metadata']['id'])
space_uid = guid_from_space_name(client,'vegetables_classification')
print("Space UID = " + space_uid)
client.set.default_space(space_uid)
client.software_specifications.list(limit=100)
software_space_uid =
client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
model_details = client.repository.store_model(model='vegetables-
model_new.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"CNN",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

```
model_details
model_id = client.repository.get_model_id(model_details)
model_id
client.repository.download(model_id,'vegetables_model_new.tar.gz')
from tensorflow.keras.models import load_model
from keras.preprocessing import image
from PIL Image
```

```
import numpy as np
model = load_model("vegetables_train.h5")
```

8.TESTING FOR VEGETABLES PREDICTION:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0


cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='o58uAinmdzXtLgQHZVw8WeFds-HPV_FiqQno1ZAFECSp',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'vegetablediseaseprediction-donotdelete-pr-tu6rk1bz8j3qu4'
object_key = 'c902f89f-a54d-44fa-8fd4-81fb9f536313___Keller.St_CG 1834.JPG'

streaming_body_6 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']
img = Image.open(streaming_body_6).convert("L")
img = img.resize( (64,64) )
img
x = np.array(img)
x=np.ones((64,64,3))
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Earl
y_blight','Potato___Late_blight','Potato___healthy','Tomato___Bacterial_spot',
'Tomato___Late_blight','Tomato___Leaf_Mold','Tomato___Septoria_leaf_spot']
preds=(index[y[0]])
preds
```

9. RESULTS

The above code will display the Predictions of the fruit and vegetable leaf diseases and also recommends the fertilizers from the excel file.

10. ADVANTAGES AND DISADVANTAGES:

Advantages:

- Prevents the spreading of disease and increases the crop field.
- User Friendly.

Disadvantages:

- Uneducated people need guidance.
- Sometimes disease prediction cannot accurate.

11. CONCLUSION:

Our project Fertilizer Recommendation System deals with identifying the various disease in plants. The main purpose of this project is to create a web page that helps the farmer to yield more crops.

12. FUTURE SCOPE:

The proposed system takes 64 x 64 pixel-sized images as input. In the future, we can build an application that can run on various OS platforms and helps in the betterment of the farmers.

13. APPENDIX::

Source Code:

Click here

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-31946-1660206833