



A Major Project Report on

**CROPS DIGITAL SOLUTIONS USING MACHINE LEARNING
AND INTERNET of THINGS**

Submitted in partial fulfillment of the requirements for the Degree of

B. Tech in Computer Engineering

by

Manik Gupta

1805311

Siddhi Vinayak Tripathi

1805219

under the guidance of

Prof. Chittaranjan Pradhan

School of Computer Engineering
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
(Deemed to be University) BHUBANESWAR

May 2022

CERTIFICATE

This is to certify that the **project (part-1)** report entitled “**CROPS DIGITAL SOLUTIONS USING MACHINE LEARNING AND INTERNET of THINGS**” submitted by

Manik Gupta

1805311

Siddhi Vinayak Tripathi

1805219

in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology** in **Computer Science Engineering** is a bonafide record of the work carried out under my guidance and supervision at the School of Computer Engineering, KIIT (Deemed to be University).

Signature of Supervisor

Prof. Chittaranjan Pradhan

School of Computer

Engineering

KIIT (Deemed to be

University)

The Project was evaluated by me on _____



ACKNOWLEDGEMENTS

In the accomplishment of this project successfully, many people have bestowed upon me their blessings and heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.

Primarily we would thank God for being able to complete this project with success. Then we would like to thank our **Dean Prof. Bhabani Shankar Prashad** and our Project mentor **Mr. Chittaranjan Pradhan**, whose valuable guidance has been the ones that helped me patch this project and make it full proof success. His suggestions and his instructions have served as the major contributor towards the completion of the project.

Also we would like to thank our parents and friends who have helped us with their valuable suggestions and guidance and have been very helpful in various phases of the completion of the project. We acknowledge here our debt to those who contributed significantly to one or more steps. We take this opportunity to express our gratitude to everyone who has been of help and assistance to us, all through the making of this project.

STUDENT SIGNATURE

Roll	Name	Signature
1805311	Manik Gupta	
1805219	Siddhi Vinayak Tripathi	

Date: -

ABSTRACT

In India, agriculture employs a lot of people, also being a major source of revenue, it becomes necessary to use our resources in a calculated manner while also making the entire farming business more profitable. Precision Agriculture is a method where resources are utilized in optimum amounts to get increased yields and profits in comparison with usual farming ways. Therefore, it is essential to develop end-to-end solutions which can help out farmers.

Although effort has been put into making farmers aware of this kind of farming method, the solutions are still incomplete and not very user-friendly for farmers to make use of. An end-to-end solution will assist the farmers in being more cautious of their decisions when it comes to crop cultivation.

TABLE OF CONTENTS

CERTIFICATE	2
ACKNOWLEDGEMENTS.....	3
ABSTRACT	4
TABLE OF CONTENTS	5
LIST OF FIGURES	6
CHAPTER 1 INTRODUCTION	7
1.1 Project Overview	7
1.2 Scope	7
1.3 Technology Used	7
1.4 Organization Of The Report	8
CHAPTER 2 DESIGN OF IoT MODULES.....	9
2.1 Module Specifications	9
2.2 System Implementation	14
CHAPTER 3 DATASETS INSIGHTS AND MODEL BUILDING	15
3.1 Crop Recommendation Dataset	15
3.2 Fertilizer Recommendation Dataset.....	21
3.3 Crop Health Prediction Dataset	22
CHAPTER 4 DEPLOYMENT OF ML MODELS ON WEBSITE BUILDING	29
4.1 Deployment Of Machine Learning Models	29
4.2 Building Website	31
4.3 Deploying Code To Herouko.....	35
CHAPTER 5 CONCLUSION	36
5.1 Conclusion.....	36
5.2 Future Scope.....	36
5.3 Cost Analysis.....	36
5.4 Project Planning And Management	37
REFERENCES.....	40

LIST OF FIGURES

Figure 1:Arduino UNO R3.....	9
Figure 2:Soil Moisture Sensor.....	10
Figure 3:Soil pH Sensor.....	10
Figure 4:DHT11 (Temperature and Humidity Sensor).....	11
Figure 5:Rainfall Sensor.....	11
Figure 6:Node MCU ESP8266.....	12
Figure 7:Jumper wires.....	12
Figure 8:Soldering Gun.....	13
Figure 9:9V Power Adapter.....	13
Figure 10:System Design for Data Reading and Transfer.....	14
Figure 11:Crop Recommendation Dataset.....	15
Figure 12:Nitrogen vs Crop Type.....	16
Figure 13:Phosphorus vs Crop Type.....	17
Figure 14:Potassium vs Crop Type.....	17
Figure 15:Temperature vs Crop Type.....	17
Figure 16:Humidity vs Crop Type.....	18
Figure 17:pH vs Crop Type.....	18
Figure 18:Rainfall vs Crop Type.....	18
Figure 19:Snippet of a Branch of a Decision Tree from the Random Forest Model.....	19
Figure 20:Confusion Matrix of Random Forest Model on test data.....	20
Figure 21: ROC-AUC Curve for the Model Performance.....	20
Figure 22:Fertilizer Recommender Dataset.....	21
Figure 23: Crop Health Prediction Dataset.....	22
Figure 24:Crop Health Dataset Description.....	23
Figure 25:Insect count vs Year Trend.....	24
Figure 26:Number of weeks quit vs Year.....	24
Figure 27:Number of weeks used vs Year.....	25
Figure 28:Soil Type vs Year.....	25
Figure 29:Counts of each Class.....	25
Figure 30: Histogram of all numerical features showing their distribution.....	26
Figure 31:Feature Importance of independent variables.....	27
Figure 32:Branch of a Decision Tree of the Random Forest Algorithm on the Crop Health Dataset.....	27
Figure 33:ROC-AUC Curve to show Model performance.....	28
Figure 34: Process of ML Model Deployment.....	29
Figure 35: Snippet of app.py rendering HTML Templates.....	30
Figure 36:Model API.....	31
Figure 37:Landing Page.....	31
Figure 38:Home Page with Nav-bar.....	Error! Bookmark not defined.
Figure 39:Responsive Nav-bar for Mobile Screens.....	32
Figure 40:Footer.....	Error! Bookmark not defined.
Figure 41:Crop Recommender Webpage.....	33
Figure 42:Fertilizer Recommender Page.....	34
Figure 43:Health Predictor Page.....	34
Figure 44:Activity Flow Diagram.....	37
Figure 45:Gantt Chart.....	39

INTRODUCTION

1.1 PROJECT OVERVIEW

In this project, we intend to solve problems in agriculture using Machine Learning and IoT. Problems like crop recommendation, fertilizer recommendation, and crop health prediction were solved using state-of-the-art supervised machine learning algorithms. For data collection, an Arduino board and 4 different sensors were used.

1.2 SCOPE

Farming is one of the major sectors that influences a country's economic growth. In a country like India, the majority of the population is dependent on agriculture for their livelihood. Precision agriculture is in trend nowadays. It helps the farmers to get informed decisions about the farming strategy.

1.3 TECHNOLOGY USED**Software Requirements: -**

1. OS – Windows 10 or higher
2. Arduino IDE
3. Visual Studio Code
4. Heroku (Platform as a Service for Deployment)

Programming Languages: -

1. Python – NumPy, Pandas, seaborn, sci-kit learn, pickle
2. Streamlit (python)– for Front-End
3. C++ - for Arduino IDE

1.4 ORGANIZATION OF THE REPORT

This project showcases how IoT and Machine Learning are used to create end-to-end solutions faced by farmers in the field of agriculture. After devising an Arduino board and using some required sensors, we were able to collect data, on which we later trained our Machine Learning Models.

The report comprises 5 chapters. Starting from Chapter 1, we have the introduction of the project which gives a brief idea about the project, its scope, and the technologies that were used in it. Chapter 2 is based on how an Arduino-based IoT circuit was connected with the to extract and create a dataset after being put in actual farms. Chapter 3 sheds some light on the dataset(s) used in our machine models and how we developed machine learning models for the same. In this chapter, it has been shown how our dataset was made usable for our machine learning model to work on, and what feature engineering methods were used to achieve that result. In Chapter 4, how a locally run machine learning model is deployed to provide real-time solutions to new and unseen data. Furthermore, making a website to interface our machine learning with has also been briefly discussed. Chapter 5 is the conclusion of the project which also includes the future scope, cost analysis of the project. It includes the project planning and management which contains the weekly planning of the project and visualizes the same with the help of a Gantt chart.

DESIGN OF IoT MODULES

To prepare our dataset on which our machine learning models work, we designed a module using NodeMCU, and various sensors.

2.1 MODULE SPECIFICATIONS

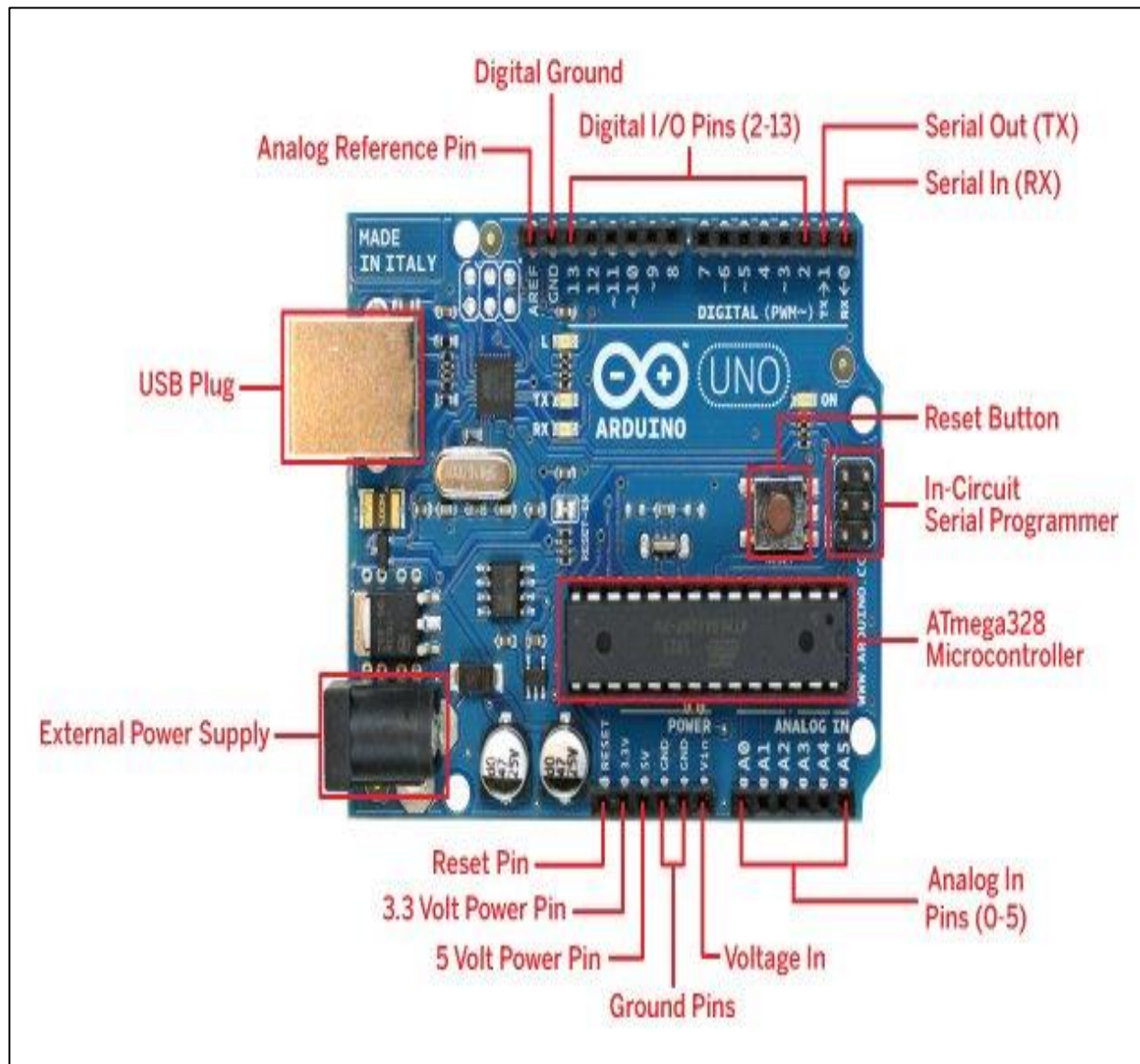


Figure 1:Arduino UNO R3

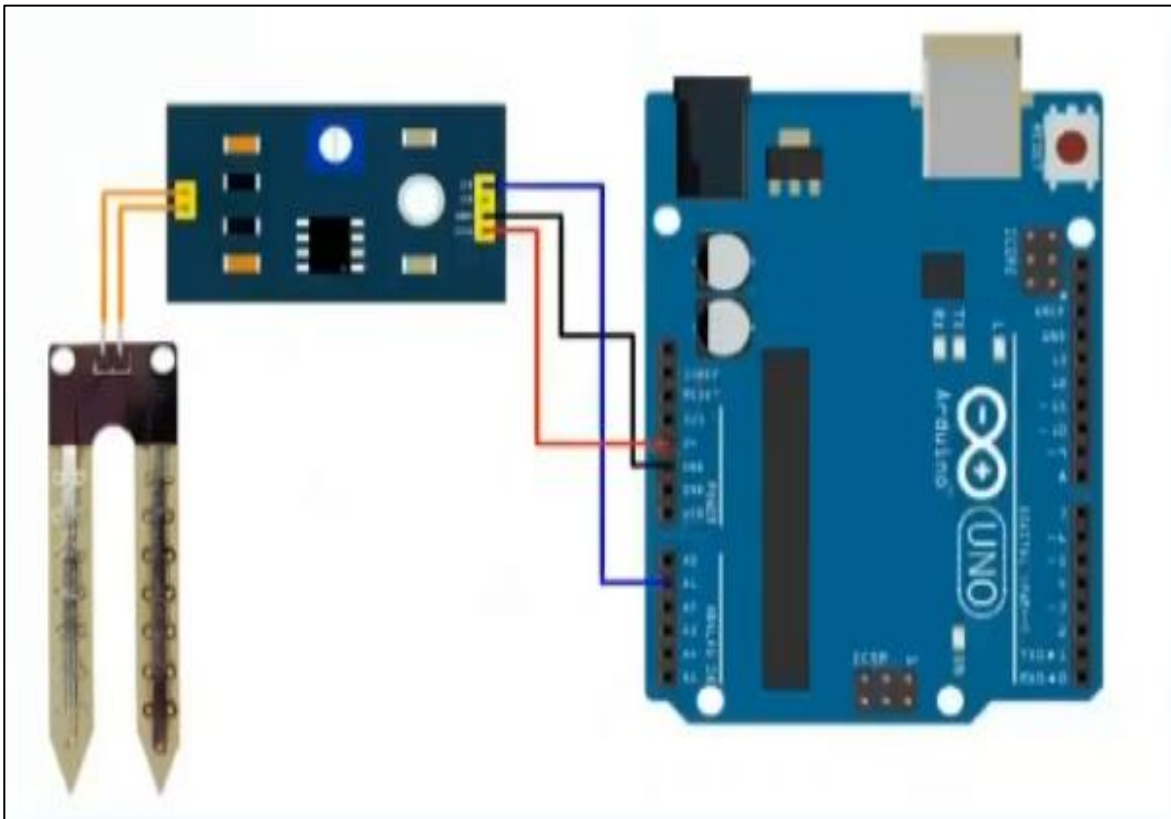


Figure 2: Soil Moisture Sensor



Figure 3: Soil pH Sensor



Figure 4:DHT11 (Temperature and Humidity Sensor)



Figure 5:Rainfall Sensor

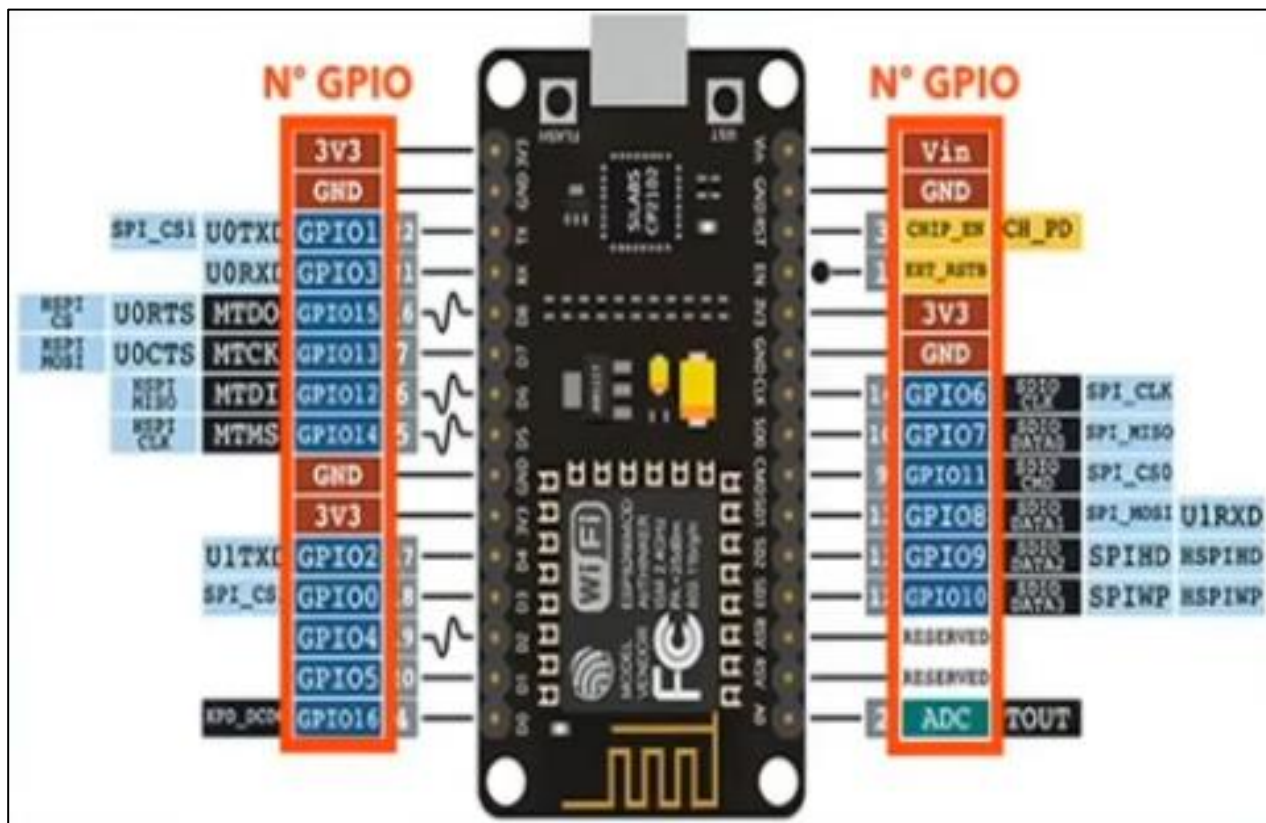


Figure 6:Node MCU ESP8266

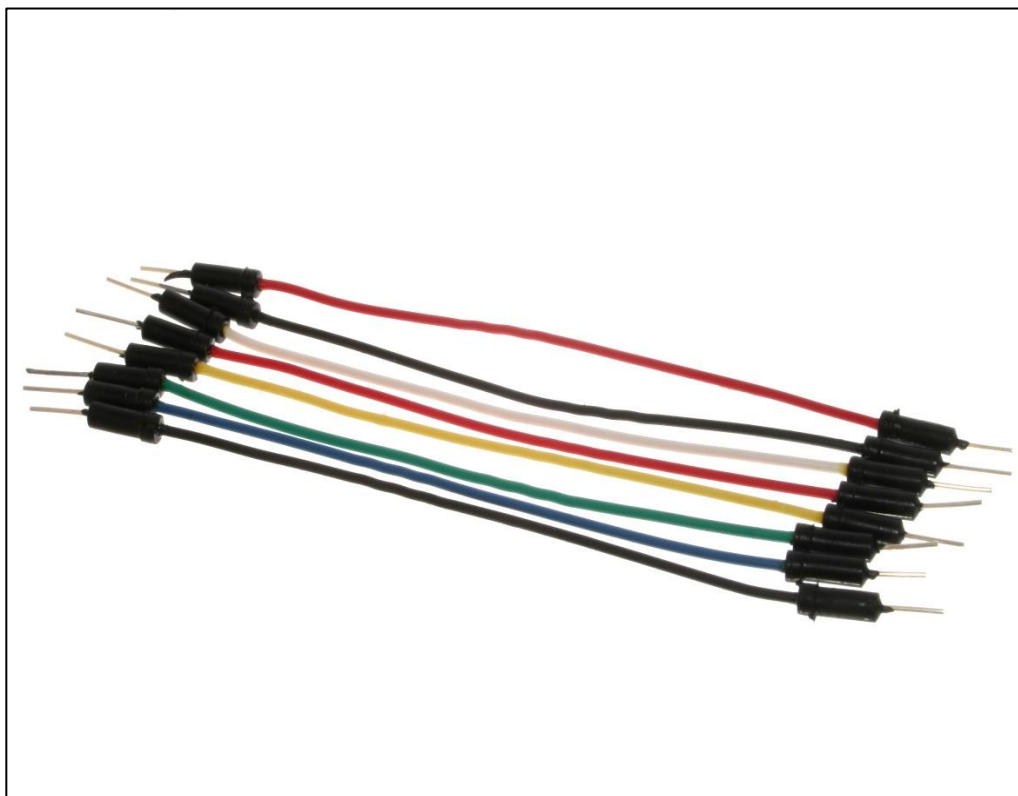


Figure 7:Jumper wires



Figure 8:Soldering Gun



Figure 9:9V Power Adapter

- ESP8266 and Arduino UNO R3 are the two boards that are used for Sensor data extraction.
- Arduino UNO is a microcontroller board based on the ATmega328P.
- The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability and is based on the ESP-12 Module.

2.2 SYSTEM IMPLEMENTATION

- Firstly, the Arduino UNO R3 connected with various Analog sensors sends the sensors data to the ESP8266(NodeMCU) Module.
- ESP8266 along with the other parameters data sends an HTTP POST request to the Apache server to store the sensor data on to the MySQL Database
- The same sensors data insertion onto the database is done repeatedly for over a period of time/seasons

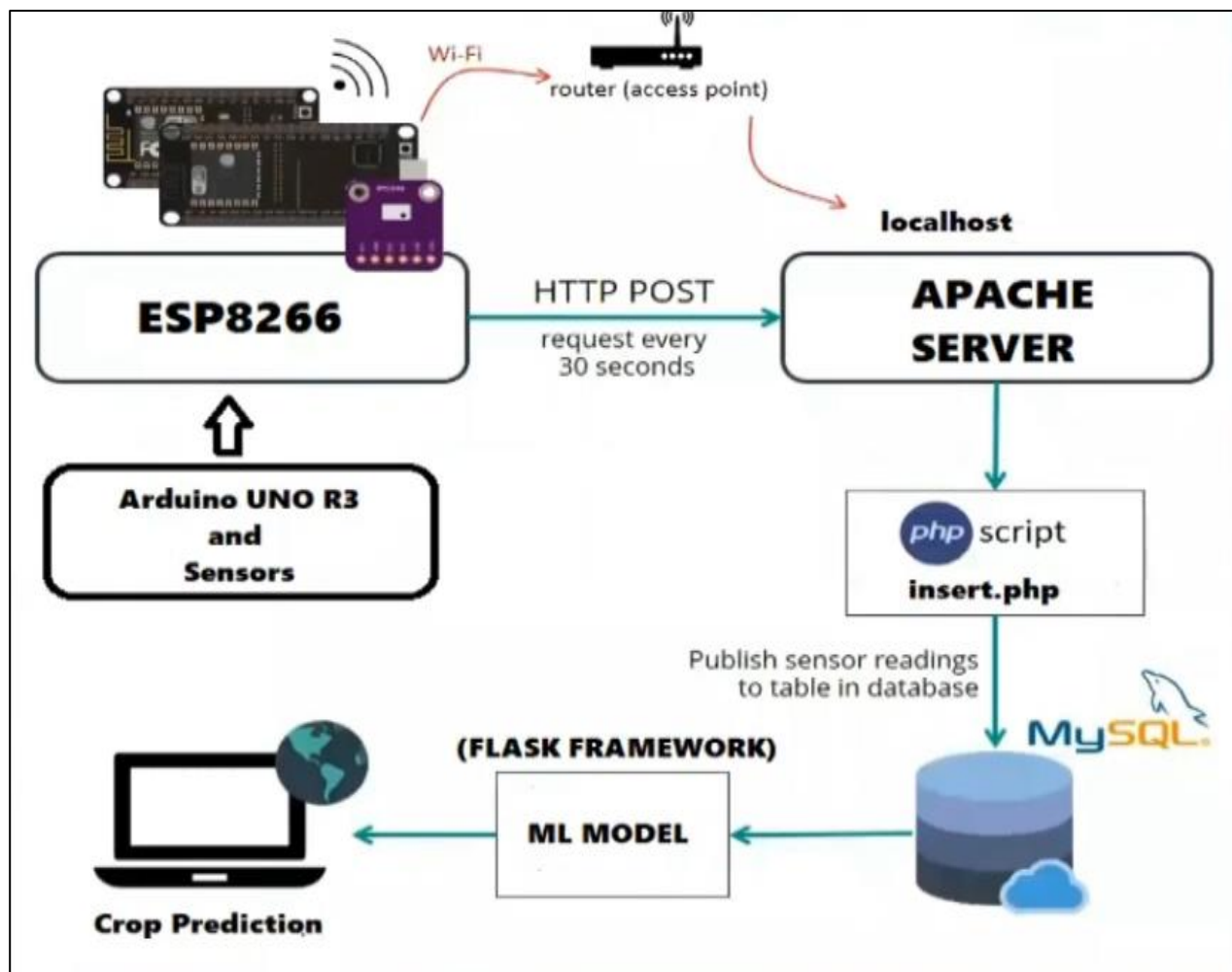


Figure 10: System Design for Data Reading and Transfer

DATASETS INSIGHTS AND MODEL BUILDING

3.1 CROP RECOMMENDATION DATASET

This dataset has been built using the data from the sensors measured in the farms. All the data was collected over a large period, from the 22 different field crops.

The data was moved from the sensors using a POST method to a MySQL server database using a PHP server.

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice
5	69	37	42	23.058049	83.370118	7.073454	251.055000	rice
6	69	55	38	22.708838	82.639414	5.700806	271.324860	rice
7	94	53	40	20.277744	82.894086	5.718627	241.974195	rice
8	89	54	38	24.515881	83.535216	6.685346	230.446236	rice
9	68	58	38	23.223974	83.033227	6.336254	221.209196	rice
10	91	53	40	26.527235	81.417538	5.386168	264.614870	rice

Figure 11: Crop Recommendation Dataset

ABOUT THE DATASET

The entire dataset comprises numerical data columns, except the last column, which is a categorical one. The numerical data in the dataset are a characteristic value of the soil that is Nitrogen Value, Phosphorus Value, Potassium Value, Temperature, Humidity, pH Value, amount of Rainfall. The values of temperature, humidity, pH, and rainfall are floating-point values.

The description of the columns in the dataset is, as follows: -

- N - the amount of Nitrogen present in the soil
- P - the amount of Phosphorus present in the soil
- K - the amount of Potassium present in the soil
- temperature - the temperature of the area in °C
- humidity – relative humidity of the farm area
- pH – aka Potential of Hydrogen depicting the acidity/basicity of the soil
- rainfall – the rainfall the area receives in millimeters

FEATURE ENGINEERING AND EXPLORATORY DATA ANALYSIS

Before building our model, we need to do some feature engineering, and data analysis so that we can better understand our data before we fit it into a model.

For Feature Engineering, we converted the categorical column of ‘label’ into a numerical one using the technique of One Hot Encoding and also removed the dummy variable trap.

In Data Analysis, we compared the dependent variable ‘label’ against all the other independent variables one by one. The trends in the data are as follows: -

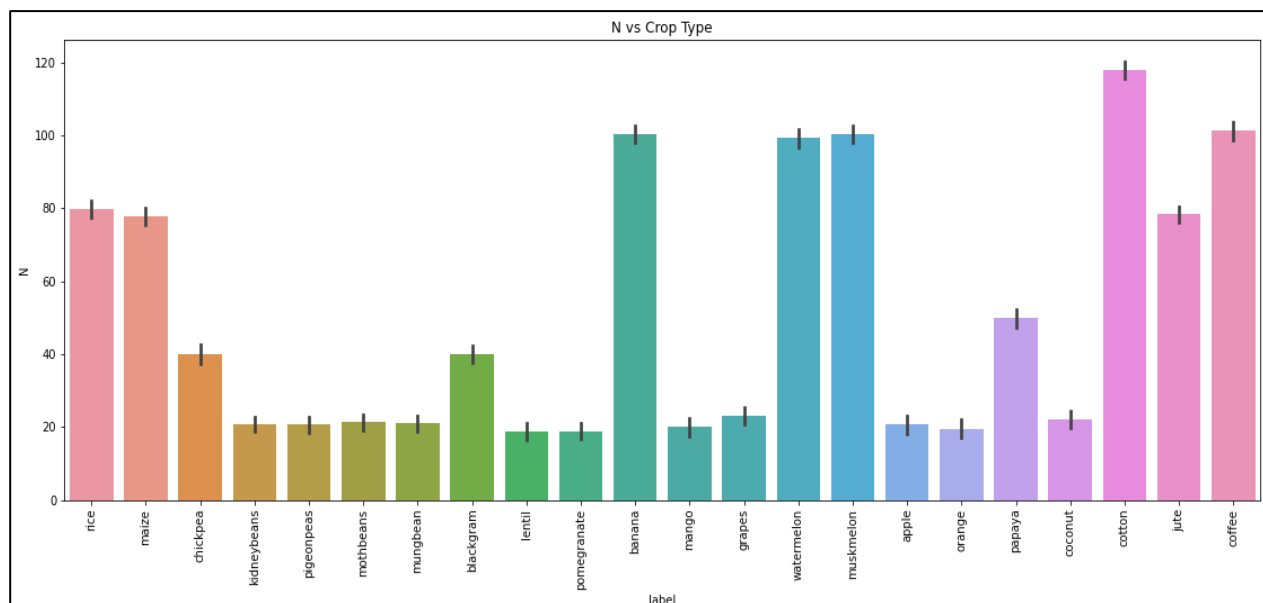


Figure 12: Nitrogen vs Crop Type

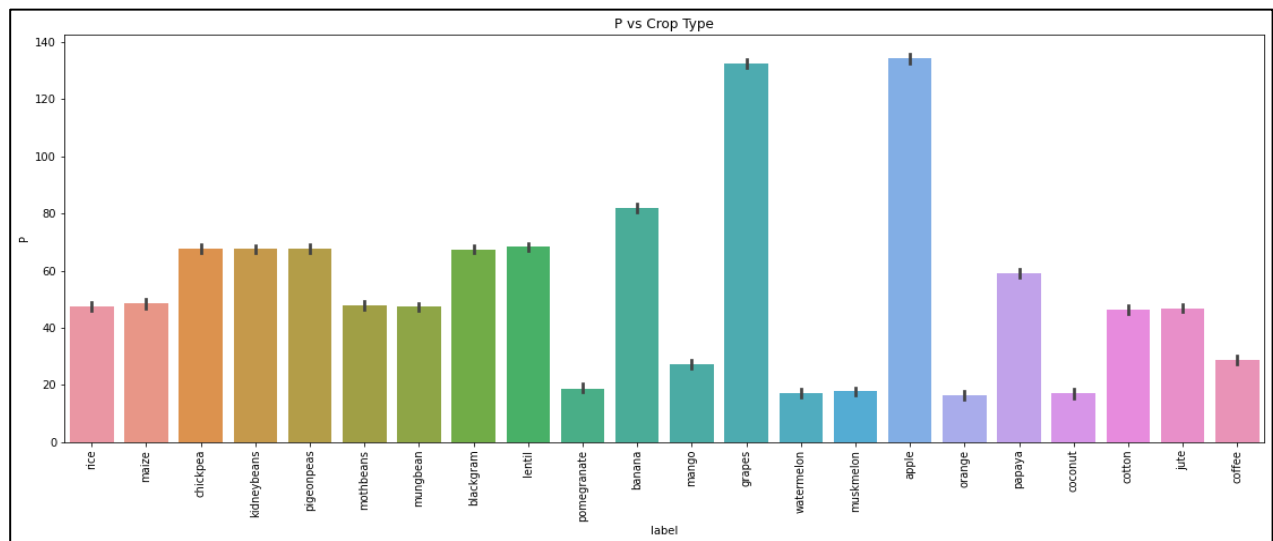


Figure 13:Phosphorus vs Crop Type

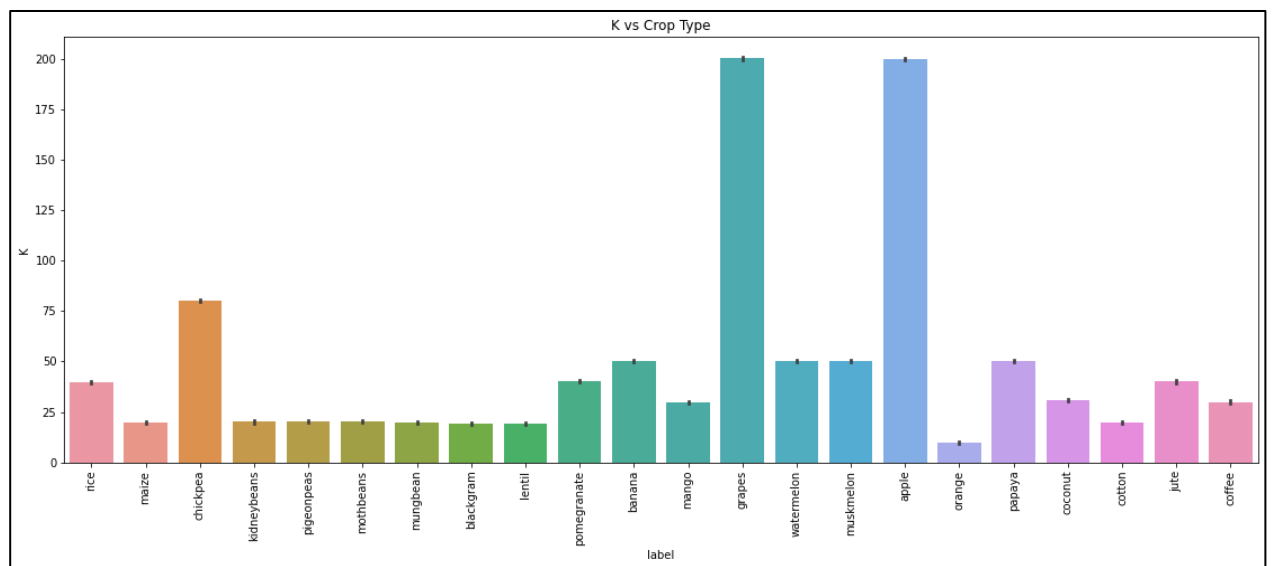


Figure 14:Potassium vs Crop Type

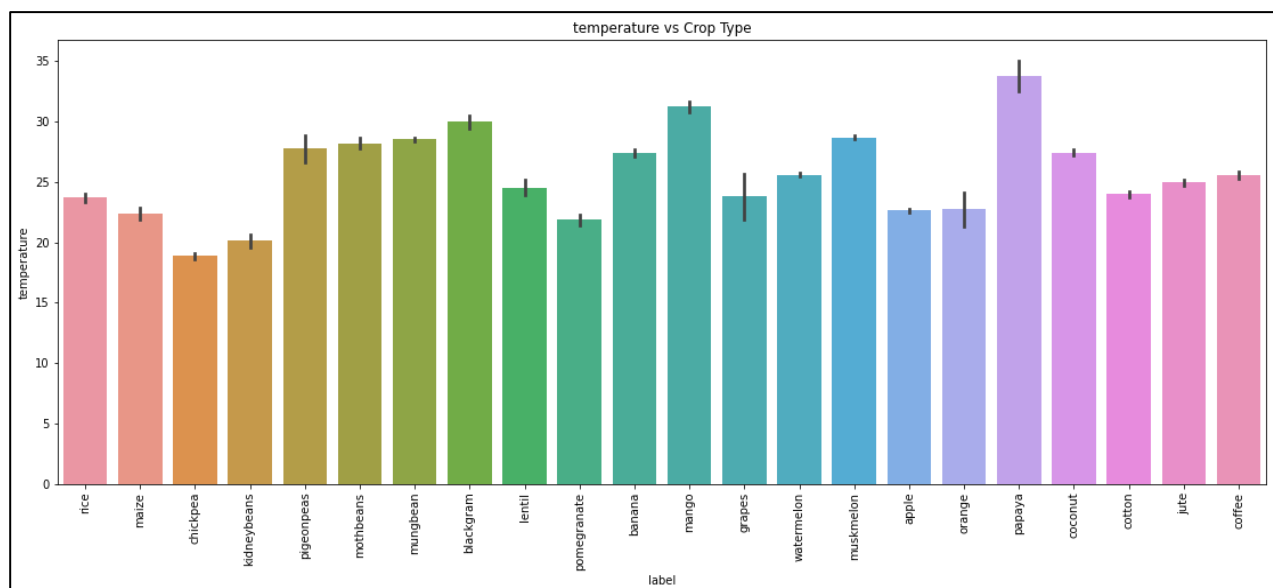


Figure 15:Temperature vs Crop Type

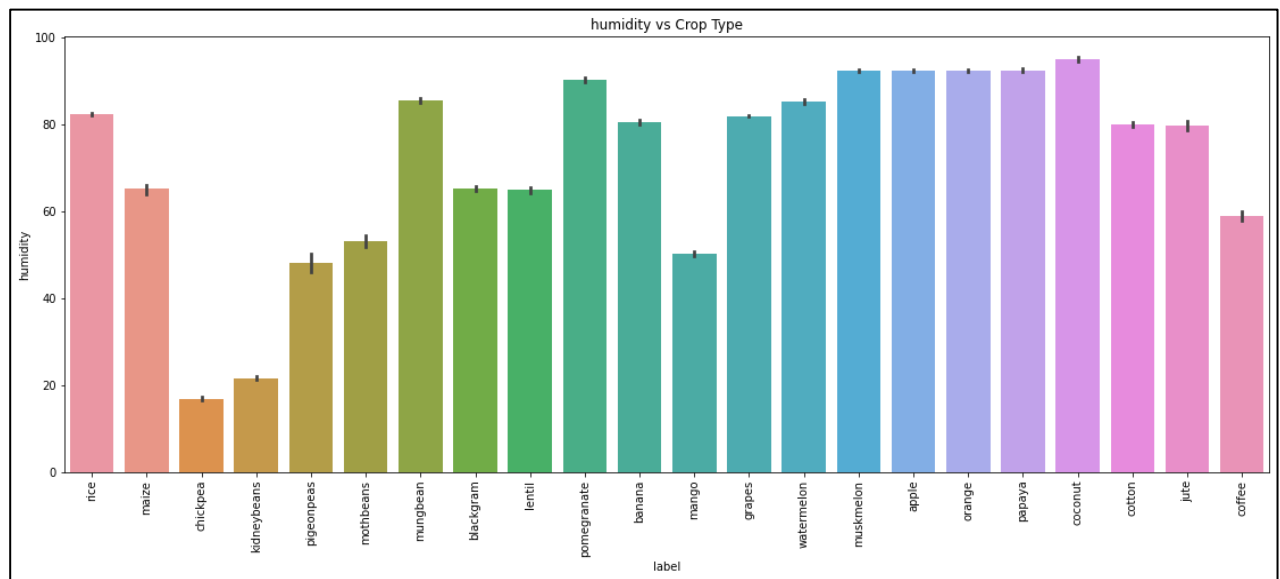


Figure 16:Humidity vs Crop Type

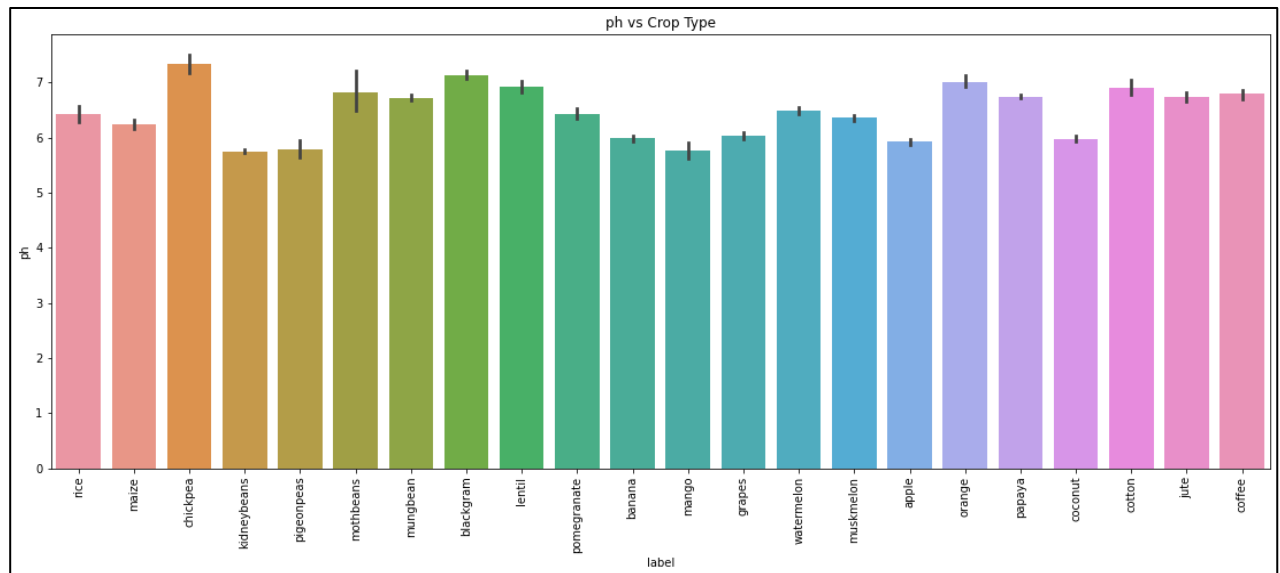


Figure 17:pH vs Crop Type

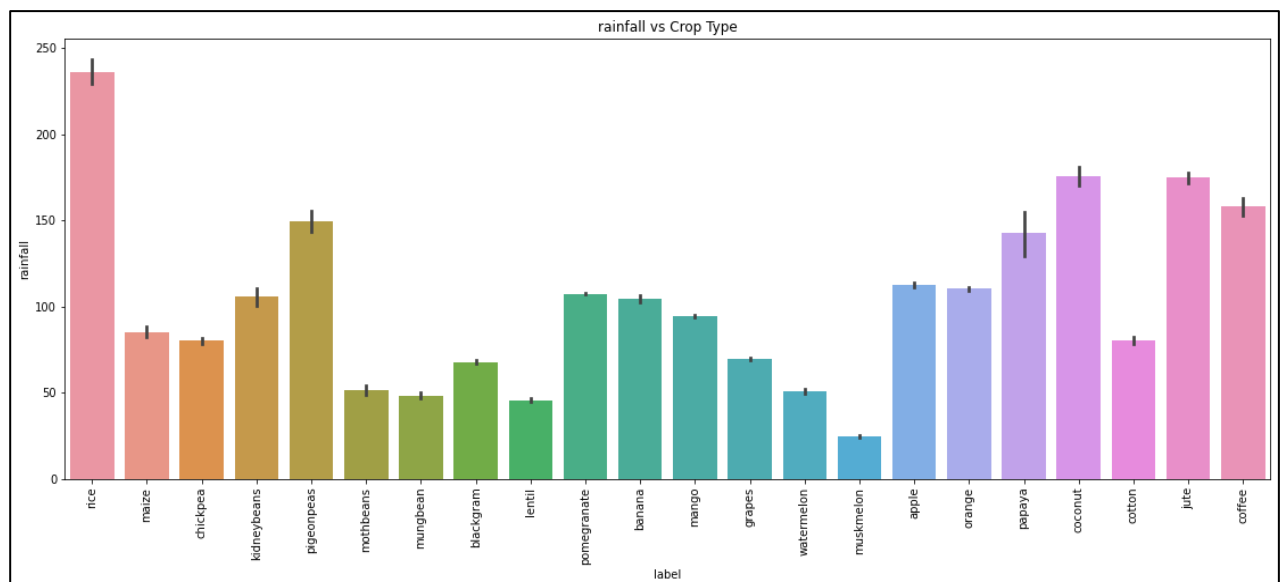


Figure 18:Rainfall vs Crop Type

MODEL BUILDING AND METRICS

We used the scikit-learn, NumPy, and panda packages for this purpose. Feature scaling was not needed as Random Forest is not a distance-based algorithm.

Figure 19: Snippet of a Branch of a Decision Tree from the Random Forest Model

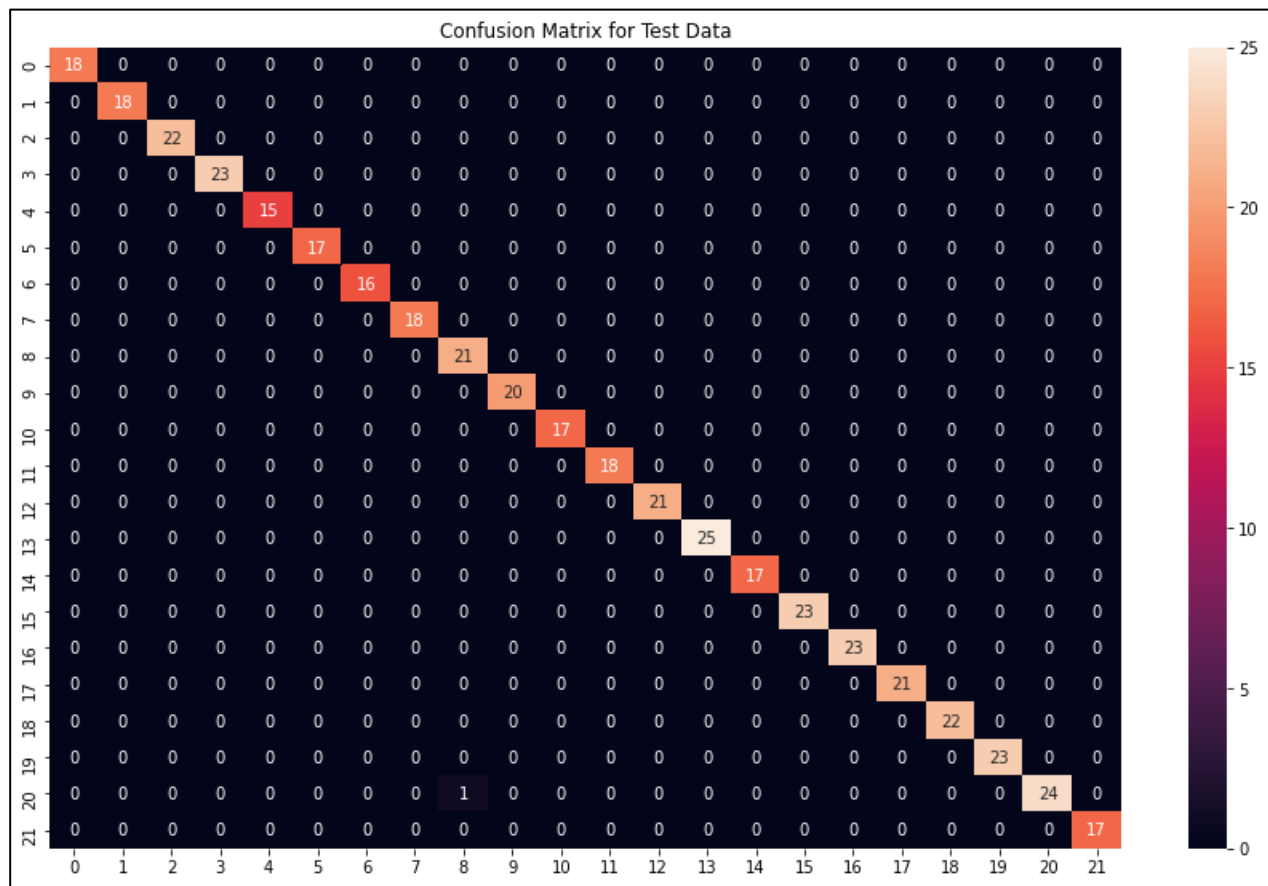


Figure 20: Confusion Matrix of Random Forest Model on test data

In the above picture, it can be seen very clearly that the Random Forest model performs well even on the unseen test data, making very few misclassifications, and thus proving again that it is the correct model for this dataset.

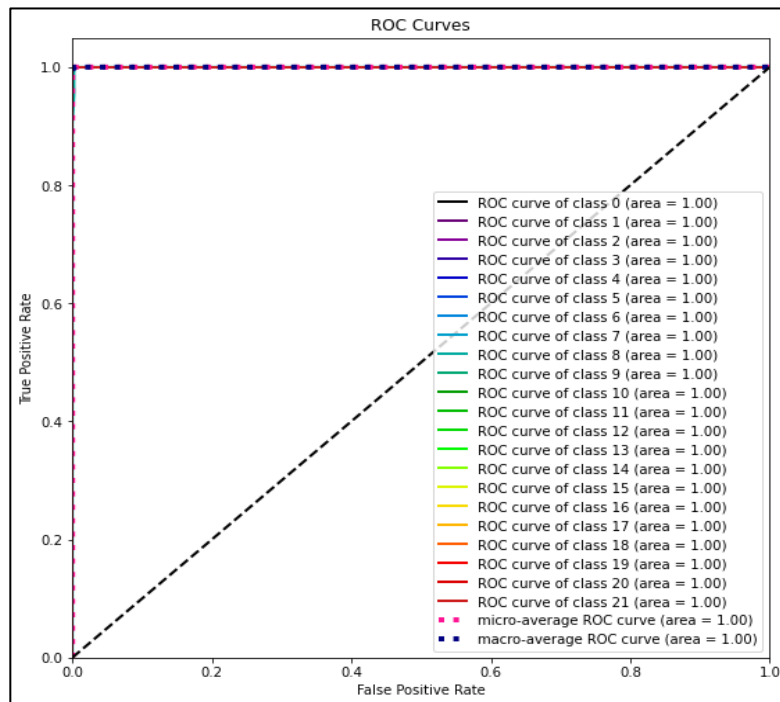


Figure 21: ROC-AUC Curve for the Model Performance

3.2 FERTILIZER RECOMMENDATION DATASET

This dataset has been used to predict which nutrients are less or more in the soil with help of the values provided to it.

	Crop	N	P	K	pH	soil_moisture
0	rice	80	40	40	5.5	30
1	maize	80	40	20	5.5	50
2	chickpea	40	60	80	5.5	60
3	kidneybeans	20	60	20	5.5	45
4	pigeonpeas	20	60	20	5.5	45
5	mothbeans	20	40	20	5.5	30
6	mungbean	20	40	20	5.5	80
7	blackgram	40	60	20	5.0	60
8	lentil	20	60	20	5.5	90
9	pomegranate	20	10	40	5.5	30
10	banana	100	75	50	6.5	40
11	mango	20	20	30	5.0	15
12	grapes	20	125	200	4.0	60
13	watermelon	100	10	50	5.5	70
14	muskmelon	100	10	50	5.5	30
15	apple	20	125	200	6.5	50
16	orange	20	10	10	4.0	60
17	papaya	50	50	50	6.0	20
18	coconut	20	10	30	5.0	45
19	cotton	120	40	20	5.5	70
20	jute	80	40	40	5.5	20
21	coffee	100	20	30	5.5	20

Figure 22:Fertilizer Recommender Dataset

ABOUT THE DATASET

The second dataset consists of categorical and numerical data. The dataset has - crop name, and their required nutrients like Nitrogen, phosphorus, potassium, and pH value, and soil moisture. All the numerical data are integer type values except the floating-point pH value. This data was also built using the data from the IoT sensors after careful study.

MODEL WORKING

For this dataset, we check the adequate values of Nutrients and pH and soil moisture of the given crop, and then we see the differences between the given value of Nutrients and soil conditions.

Our code comprises of if-else conditional statements rather than sci-kit learn-based ML models.

3.3 CROP HEALTH PREDICTION DATASET

This dataset helps to predict the health of our crop based on various conditions. Crop Damage – ‘0’ signifies no damage, ‘1’ signifies damaged crop but not from pesticide use, ‘2’ signifies damaged crop due to pesticide use.

Estimated_Insects_Count	Crop_Type	Soil_Type	Pesticide_Use_Category	Number_Doses_Week	Number_Weeks_Used	Number_Weeks_Quit	Season	Crop_Damage
188	1	0	1	0	0.0	0	1	0
209	1	0	1	0	0.0	0	2	1
257	1	0	1	0	0.0	0	2	1
257	1	1	1	0	0.0	0	2	1
342	1	0	1	0	0.0	0	2	1
448	0	1	1	0	NaN	0	2	1
448	0	1	1	0	NaN	0	2	1
577	1	0	1	0	0.0	0	1	2
731	0	0	1	0	0.0	0	2	0
1132	1	0	1	0	0.0	0	1	2
1212	1	0	1	0	NaN	0	3	0
1575	0	0	1	0	0.0	0	1	1
1575	0	1	1	0	0.0	0	2	1
1575	1	1	1	0	0.0	0	2	1
1575	1	1	1	0	0.0	0	2	2

Figure 23: Crop Health Prediction Dataset

ABOUT THE DATASET

The description for the various columns in this dataset is as follows:

Variable	Definition
ID	UniqueID
Estimated_Insects_Count	Estimated insects count per square meter
Crop_Type	Category of Crop(0,1)
Soil_Type	Category of Soil (0,1)
Pesticide_Use_Category	Type of pesticides uses (1- Never, 2-Previously Used, 3-Currently Using)
Number_Doses_Week	Number of doses per week
Number_Weeks_Used	Number of weeks used
Number_Weeks_Quit	Number of weeks quit
Season	Season Category (1,2,3)
Crop_Damage(Target)	Crop Damage Category (0=alive, 1=Damage due to other causes, 2=Damage due to Pesticides)

Figure 24:Crop Health Dataset Description

Much of the data here is categorical data, except a few which are numerical. All values in features columns are integer type values. The target of this data is to predict the health of our crop whether it is alive, damaged by other reasons, or damage by the usage of pesticides. The major affecting columns can be seen above in the figure very clearly

FEATURE ENGINEERING AND DATA ANALYSIS

The test/training datasets were combined first of all and were ordered using their ID. The very first trend encountered, was the repetitive pattern of estimated_insect_count.

We require a column that would change as soon as strictly increasing function drops. This will help the model learn better. Insect counts rise to a value of 80K to 85K then decrease.

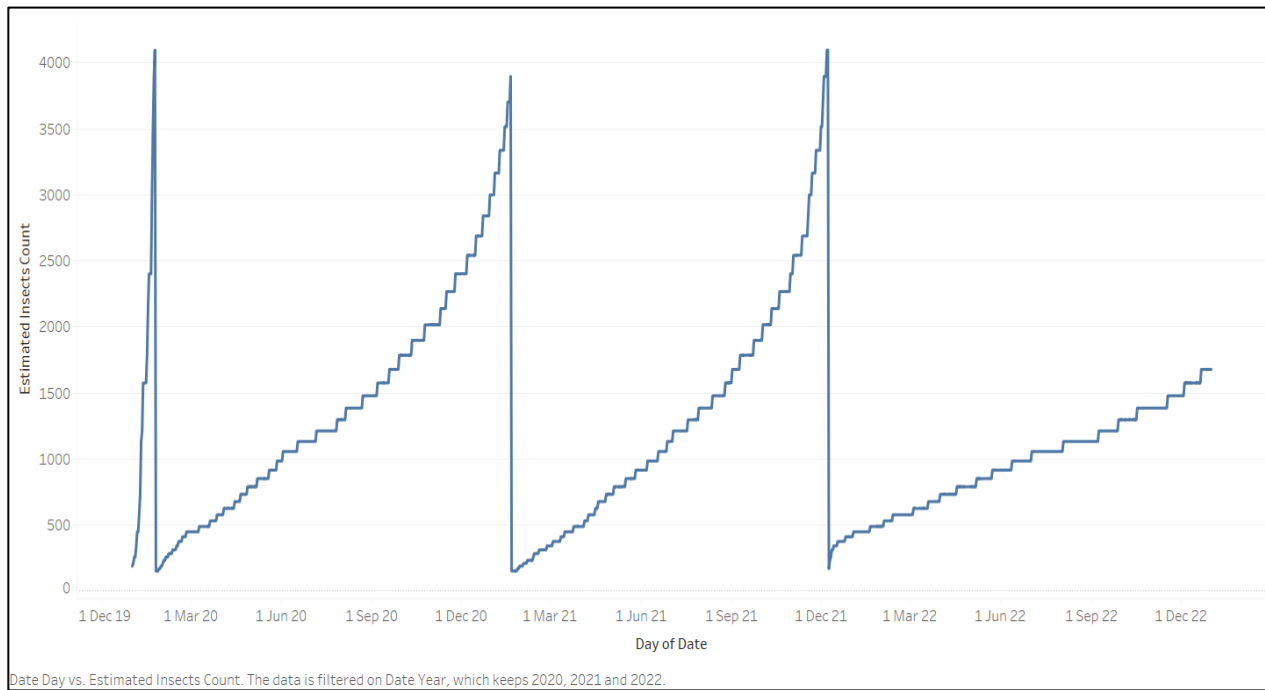


Figure 25: Insect count vs Year Trend

A very similar pattern was found within the number_of_weeks_Quit. It rises to around 45 and then drops suddenly.

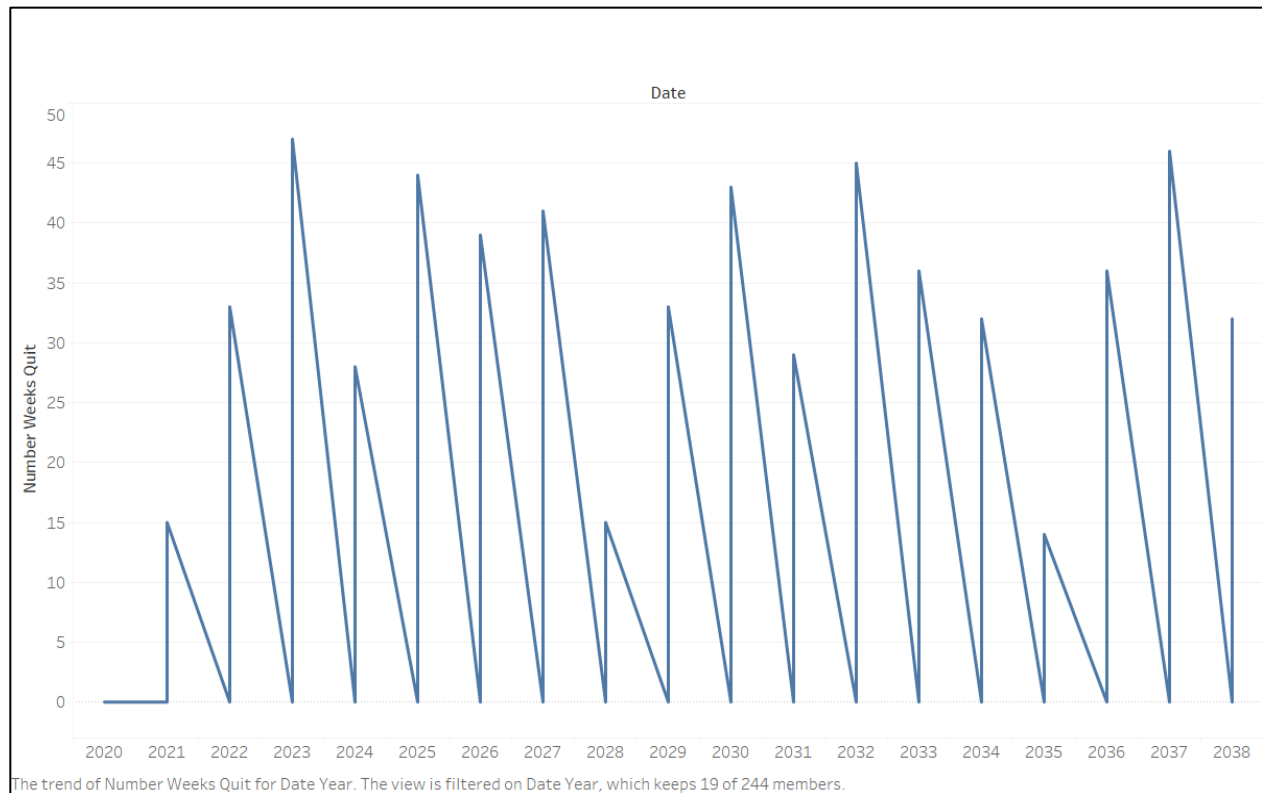


Figure 26: Number of weeks quit vs Year

Again, the same pattern with number_of_weeks_used was seen.

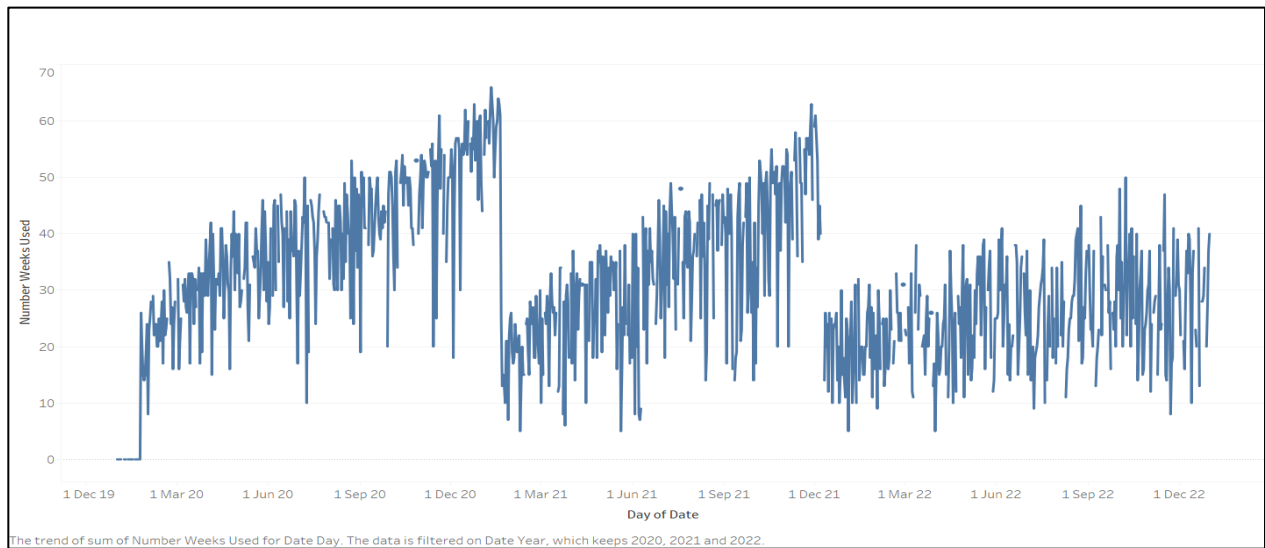


Figure 27: Number of weeks used vs Year

Similarly, soil_type also pertained to this increasing/decreasing trend.

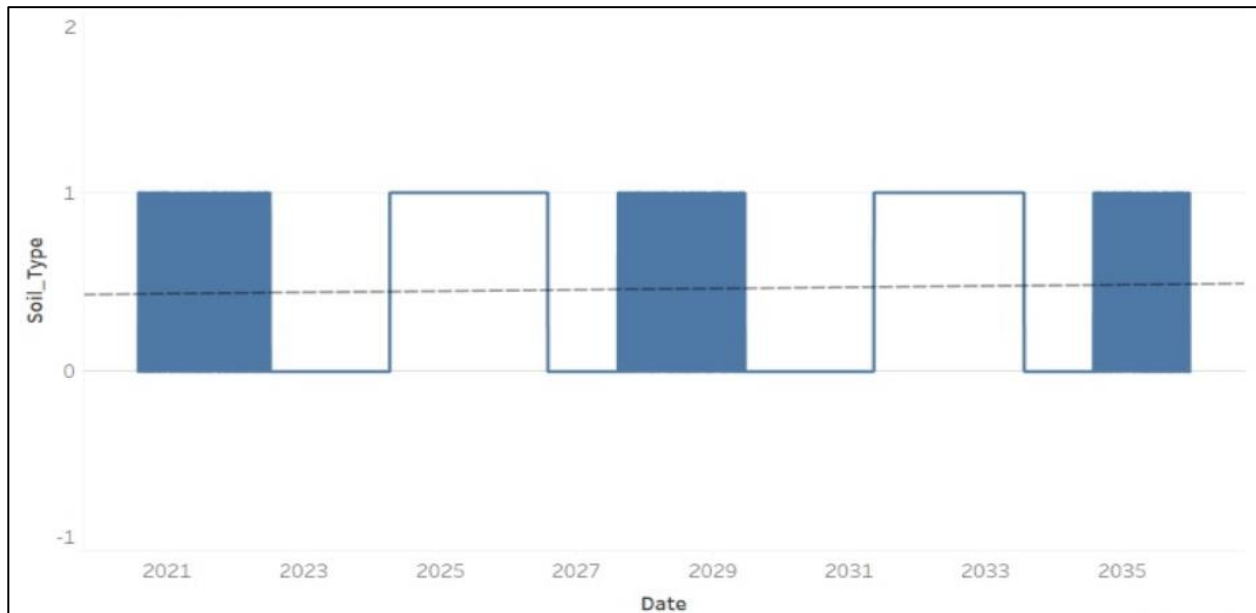


Figure 28: Soil Type vs Year

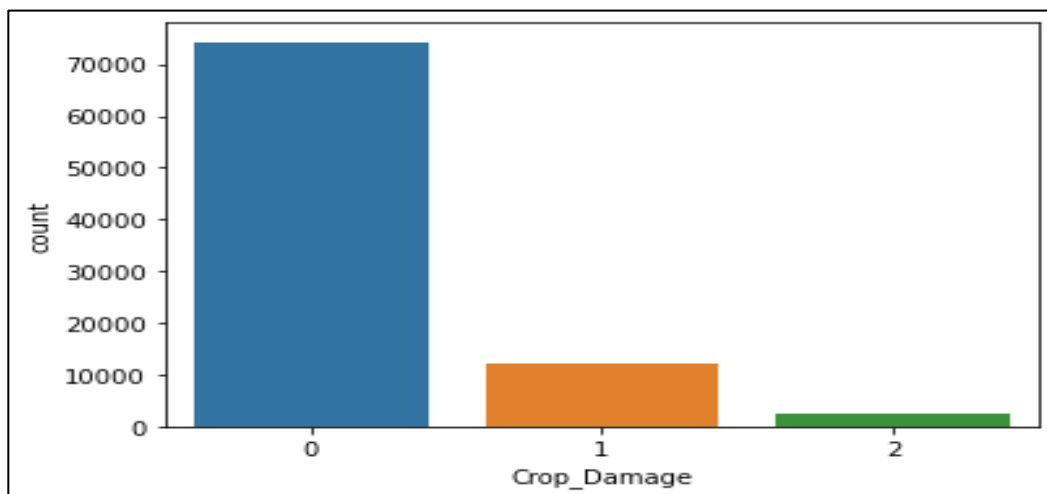


Figure 29: Counts of each Class

As it can be seen here, there was a huge class imbalance in the data, as counts of severely damaged crops due to pesticides were very less as compared to counts of healthy crops.

To make this dataset ready for the model we removed the column- id and we also imputed the missing values in the column – number_weeks_used, with its mean value. Furthermore, we analyzed the distribution of all the numerical value columns.

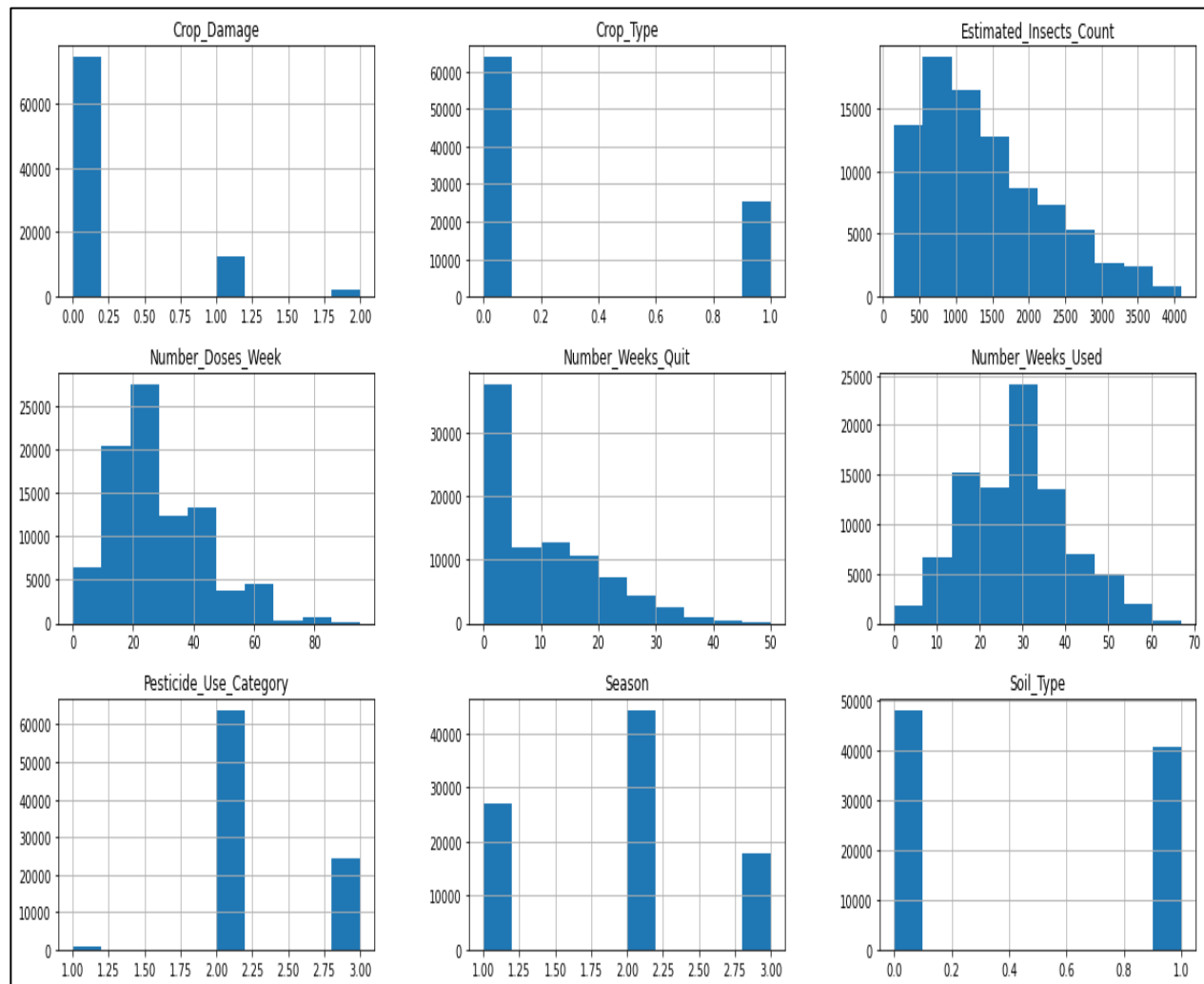
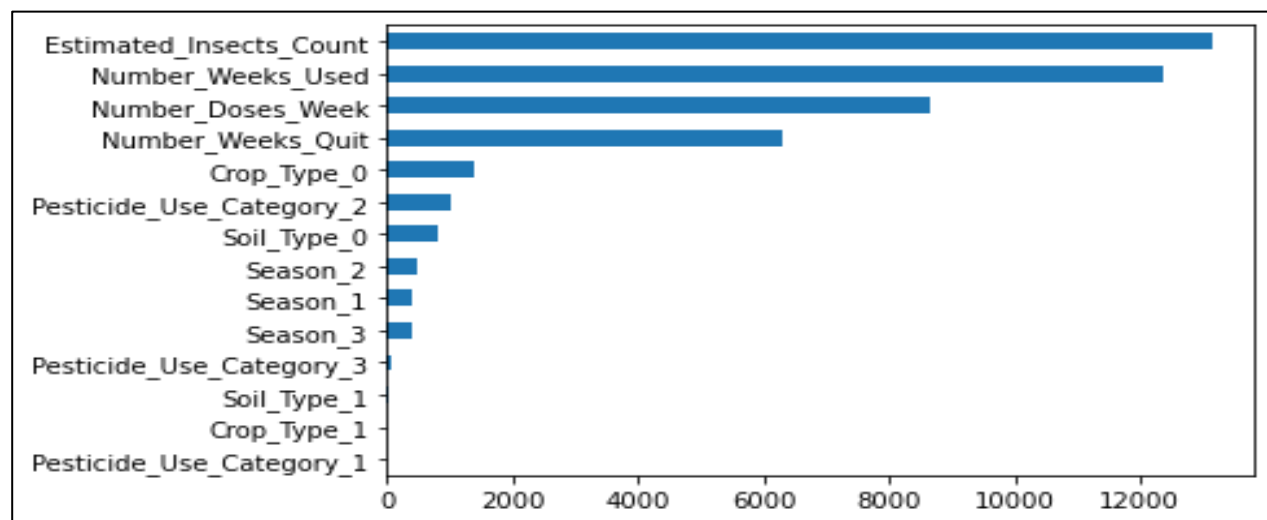


Figure 30: Histogram of all numerical features showing their distribution

It can be seen clearly that some of the above columns were skewed and they needed to be fixed. Furthermore, dummy variable encoding was also done to turn the categorical variables into numerical ones.

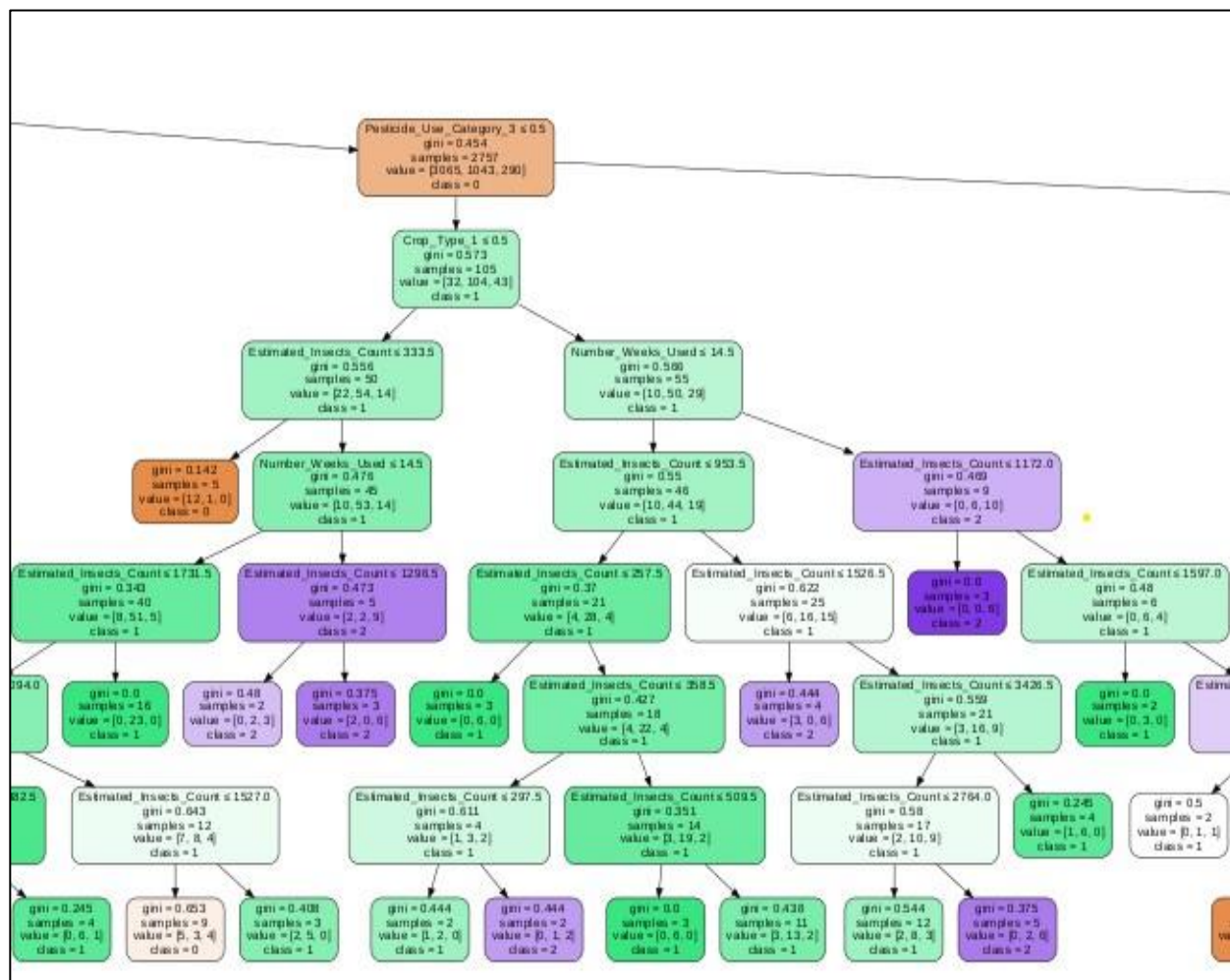
We also did some feature selection, to see which independent features were important for the prediction and which weren't.



MODEL BUILDING AND METRICS

In this dataset also, we found that the random forest algorithm fitted well on our data with little to no overfitting. A lot of parameter tuning was done with the help of the GridSearchCV library.

The resultant Random Forest can be seen below.



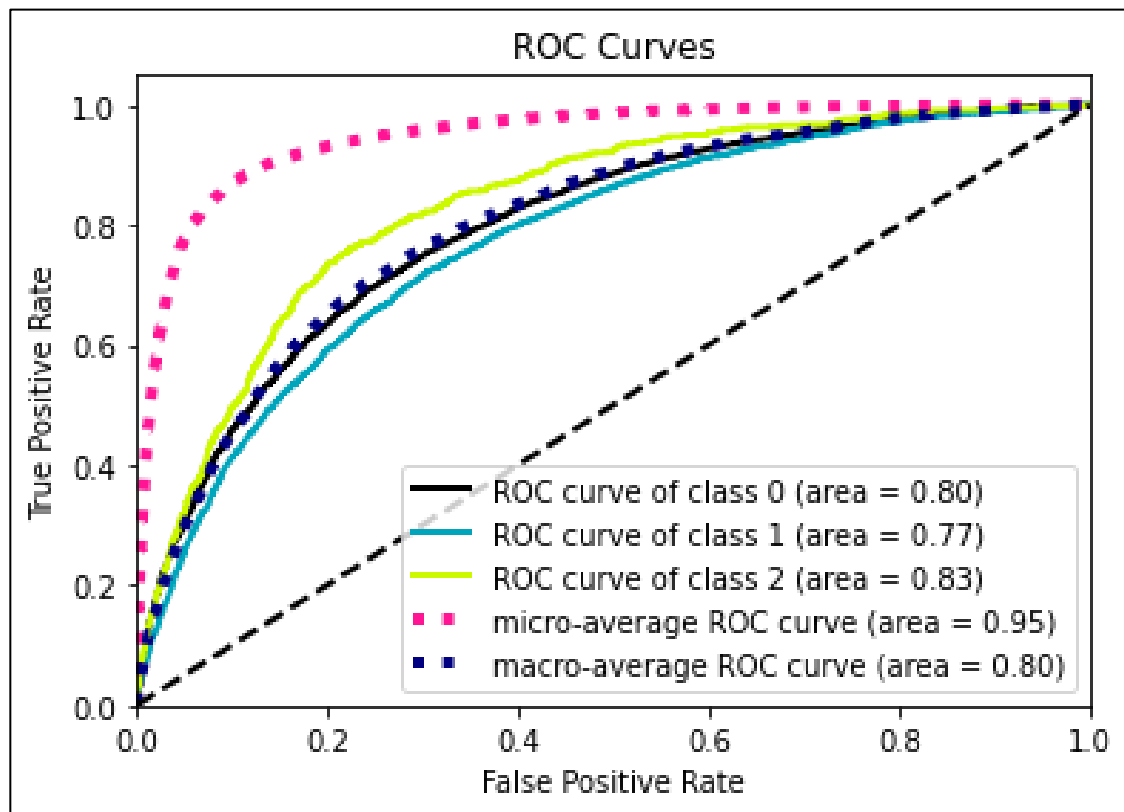


Figure 33:ROC-AUC Curve to show Model performance

The model can be seen performing well. It seems to be doing a good job of fishing out the class-2 type data from the rest of them, which in this scenario is very important. It isn't very effective in recognizing the class-1 type data, which shows more scope of improvement.

DEPLOYMENT OF ML MODELS ON WEBSITE BUILDING

4.1 DEPLOYMENT OF MACHINE LEARNING MODELS

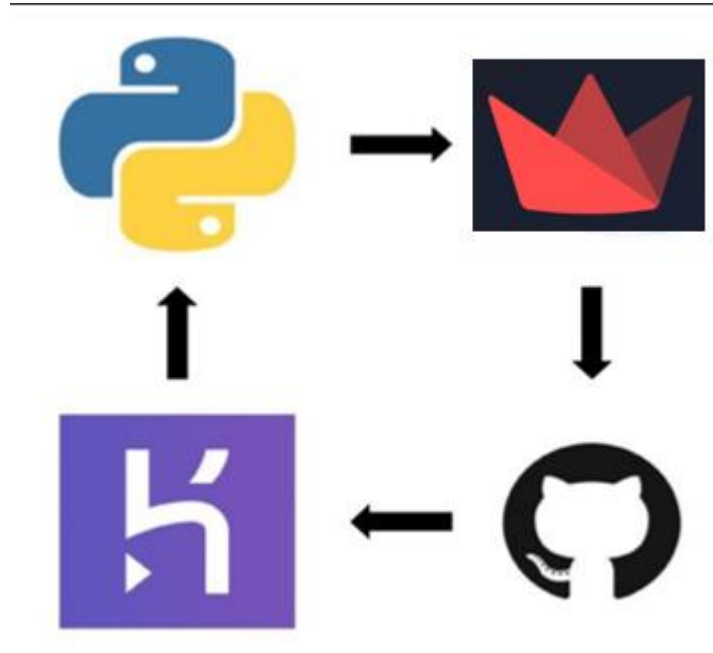


Figure 34: Process of ML Model Deployment

For model deployment, we require the following things:

- Trained Machine Learning model packaged in a ‘. pkl’ file using the pickle library.
- Web app integrated with the trained model.
- Deployment platforms like Google Cloud, Azure, AWS, Heroku.

Heroku provides a free deployment facility whereas Azure and AWS are expensive.

To deploy the machine learning model on the Herouko platform, an API will be created. API stands for Application Programming Interface. It is used to access features or data specific to the application. There are mainly two files in the API. The first is the machine learning model which evaluates the input data and the second is the application which is the part of the API which receives the input and returns the output.

```

import streamlit as st
import time
import pickle
import pandas as pd
import numpy as np

> with st.sidebar: ...

> if service=='Crop': ...

> elif service=='Fertilizer': ...

> elif service=='Health':|...

```

Figure 35: Snippet of cropDashboard.py rendering streamlit widgets

The application starts by importing the necessary libraries and then creates a Streamlit Dashboard. The machine learning model is stored in the pickle file which is then loaded into the variable model. With the help of sidebar the service is chosen and models are rendered accordingly. This approach is used to retrieve input information.

The machine learning model begins by importing the necessary libraries, similar to the application API. The model then extracts the data that will be used to test and train the model. Since this data is a CSV, it is transformed into a NumPy array. The model then formats the data a bit more so that it can be used in the model. The X and Y values are defined and transformed into integers. Training and test data are separate. The model is then trained and tested using Random Forest.

Now that the model has been trained and tested, it is loaded into a pickle file. This file is used to store the model so that it can be used in the API application. Then the following code is executed using the following command.

\

```

if submitted:
    if nitrogen and phosphorus and potassium and temperature and humidity and ph and rainfall:
        st.success("Submitted")
        model = pickle.load(open('model.pkl', 'rb'))
        final_features = np.array([[int(nitrogen), int(phosphorus), int(potassium), float(temperature), float(humidity), int(ph), int(rainfall))])
        prediction = model.predict(final_features)
        output = prediction[0]
        st.write(output)
        st.stop()
    else:
        st.error("All feilds are mandatory")

```

Figure 36:Model API

The above file is an image of the cropDashboard.py file. Each service has its own submit button and its way of dealing with the mandatory inputs and feeding it to the model. Once the model returns the prediction/recommendation we write the output to the dashboard.

4.2 BUILDING WEBSITE

The dashboard lets you chose between three services provided Crop service being the default. Each service has its own dashboard screen and form. The web pages are responsive to accommodate various screen sizes.

Figure 37:Landing Page

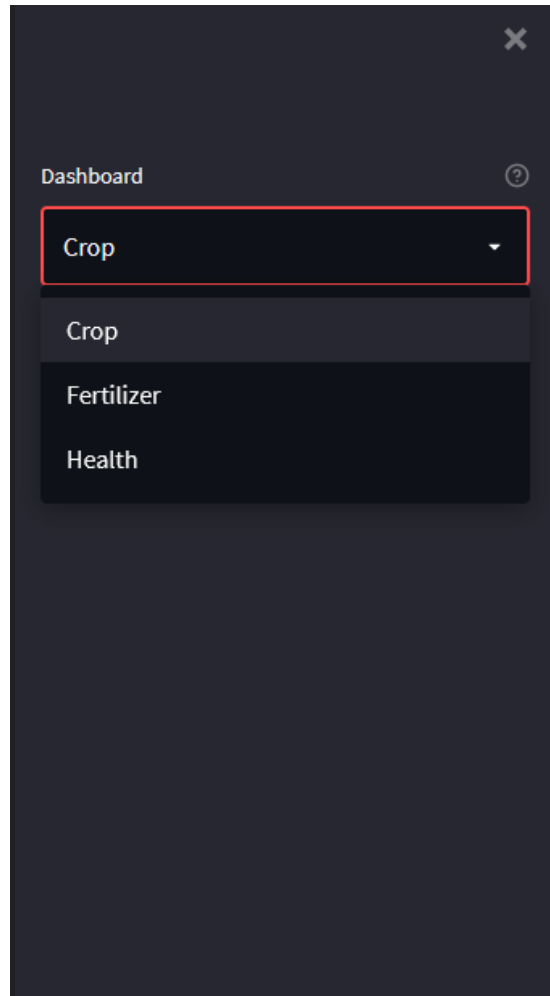


Figure 38: Responsive Nav-bar for Mobile Screens

Find out the best crop for your farm

Nitrogen

0-140

Phosphorus

5-145

Potassium

5-205

Temperature

value (in C)

Humidity

14.0-100

pH

0-14

Rainfall

0-300

Recommend

Figure 39: Crop Recommender Webpage

Dashboard

Fertilizer

Done!!

Find out the best Fertiliser for your Crop

Crop

rice

Nitrogen

0-140

Phosphorus

5-145

Potassium

5-205

Soil Moisture

Moisture Content Value

Suggest

Made with Streamlit

Figure 39: Fertilizer Recommender Page

Dashboard

Health

Done!!

Find out about your crop's Health

Insect Count

Insect Count Value

Crop Type

0 1

Soil Type

0 1

Pesticide Category

1 2 3

Season

1 2 3

Doses Per Week

Number of Doses per week

Weeks Used

Number of Weeks Used

Weeks Quit

Number of Weeks Quit

Predict

Made with Streamlit

Figure 41: Health Predictor Page

The UI of these pages has been built using Streamlit

4.3 DEPLOYING CODE TO HEROKU

Now we have the API and app in different folders of a GitHub repository by now. We then create a Heroku account to get started.

Once the app is created, we can choose our deployment method. Since our code is already uploaded to a GitHub repository, we can connect Heroku to GitHub and select the repository we want.

We can deploy the code by choosing the deploy branch option beneath the deployment method.

After the app is deployed, we can access it online by pressing open app in the top right corner.

CONCLUSION

5.1 CONCLUSION

It perfectly meets the system's goals and requirements. The IoT module was successful in fetching the data from the sensors and making a dataset. The machine learning models perform well, with an accuracy of 99.7% and 83.4%. Errors in the models after extensive feature engineering. We have proposed machine learning algorithms backed by scientific explanations to recommend crops, fertilizers and estimate health.

5.2 FUTURE SCOPE

A lot more ML Models can be designed and deployed to this site, to make this website a one-stop solution for farmers and agriculture enthusiasts. A couple of these ideas are – crop price predictor, leaf disease detector.

5.3 COST ANALYSIS

Since the project work is based on simulation and prediction, there is no actual procurement cost required.

5.4 PROJECT PLANNING AND MANAGEMENT

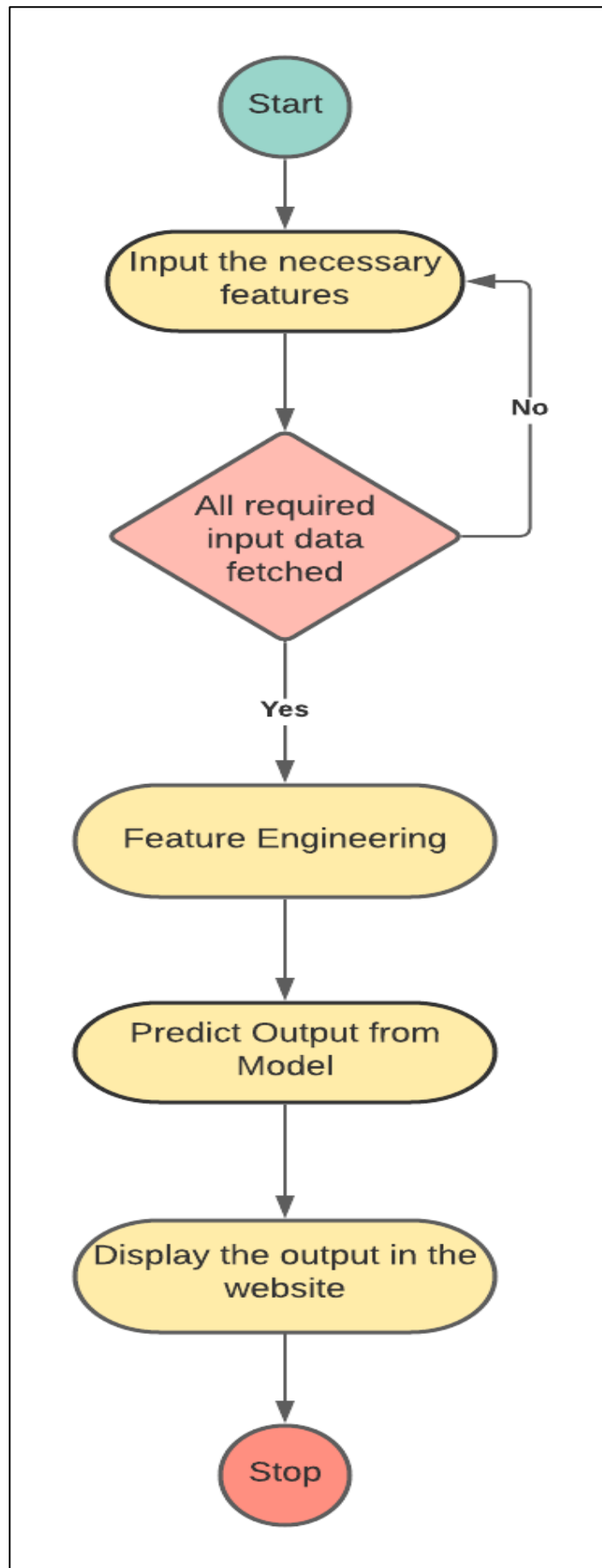


Figure 40:Activity Flow Diagram

Activity	Starting week	Weeks
Ideation	3 rd week of January	0.5
Installation Dependencies	3 rd week of January	0.5
Data Gathering, Preparation & Analysis	1 st week of February	2
Feature Engineering	3 rd week of February	2
Model Building & Evaluation	2 nd week of March	2
Dashboard Design	4 th week of March	1
Streamlit dashboard Creation	1 st week of April	1
Deployment on Heroku	2 nd week of April	1
Project Presentation & Report	3 rd week of April	1

Table 1: Details of Project Planning and Management

The Gantt chart is shown below –



Figure 41:Gantt Chart

REFERENCES

- [1] S. Shaikh, G. Dhuri, H. Mhetre, P. Raut, and A. Dhakne, "Soil Ph Value Prediction and Crop Recommendation", *International Journal of Research and Analytical Reviews*, vol. 5, no.4, (2018), pp. 670-673.
- [2] K. Abhang, S. Chaughule, P. Chavan, S. Ganjave, "Soil Analysis and Crop Fertility Prediction", *International Research Journal of Engineering and Technology*, vol. 5, no. 03, (2018), pp. 3106-3108
- [3] Y. Wang, T. Huang, J. Liu, Z. Lin, S. Li, R. Wang, and Y. Ge, "Soil pH value, organic matter and macronutrients contents prediction using optical diffuse reflectance spectroscopy", *Computers and Electronics in Agriculture*, vol. 111, (2015), pp.69-77.
- [4] R. Ghadge, J. Kulkarni, P. More, S. Nene, R. L. Priya, "Prediction of Crop Yield Using Machine Learning", *International Research Journal of Engineering and Technology*, vol. 5 no. 2, (2018), pp. 2237- 2239.
- [5] Z. Libohova, C. Seybold, K. Adhikari, S. Wills, D. Beaudette, S. Peaslee, P.R. Owens, "The anatomy of uncertainty for soil pH measurements and predictions: Implications for modelers and practitioners", *European Journal of soil science*, vol. 70, no. 1, (2019), pp.185-199.
- [6] F. Zeng, S. Ali, H. Zhang, Y. Ouyang, B. Qiu, F.Wu, G. Zhang, "The influence of pH and organic matter content in paddy soil on heavy metal availability and their uptake by rice plants", *Environmental Pollution*, vol. 159, no. 1, (2011), pp.84-91.

DIGITAL FARMING SOLUTIONS USING MACHINE LEARNING AND INTERNET of THINGS

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

2%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

www.coursehero.com

Internet Source

4%

2

Submitted to University of Ulster

Student Paper

1%

3

www.ijareeie.com

Internet Source

1%

4

Submitted to The Hong Kong Polytechnic University

Student Paper

1%

5

Submitted to Intercollege

Student Paper

<1%

6

Submitted to University of Liverpool

Student Paper

<1%

7

Deepti Lourembam, Shakuntala Laskar, Subra Mukherjee. "Framework for an optical sensor system for monitoring of soil nitrogen and tailoring soil pH", Journal of Optics, 2017

Publication

<1%



T. Malapane, W. Doorsamy, B.S. Paul. "An Intelligent IoT-based Health Monitoring System", 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), 2020

Publication

<1%

Exclude quotes On

Exclude matches < 10 words

Exclude bibliography On