HARVEST HELPERS USING AI APPROACHES TO OPTIMIZE NUTRIENT USAGE

A PROJECT REPORT

Submitted by

THRISHA M (2116210701292)

VAMSEE RAJ MR (2116210701300)

YAMINI H (2116210701320)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Thesis titled **"HARVEST HELPERS USING AI APPROACHES TO OPTIMIZE NUTRIENT USAGE**" is the bonafide work of "**THRISHA M (2116210701292), VAMSEE RAJ MR (2116210701300), YAMINI H (2116210701320)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Shanmugam M.E., Ph.D.,

Associate Professor

PROJECT COORDINATOR

Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on__________________

Internal Examiner **External Examiner**

ABSTRACT

Fertilizer use is typically under the limited control of farmers. For the farmers to achieve higher yields and reduce fertilizer loss, competent guidance is required for the best use of these fertilizers. Additionally, there is a connection between rainfall volume and nutrient loss for various fertilizer applications after each rainfall event. Rainfall that is moderate and falls at the right moment can help nutrients penetrate the soil's rooting zone and dissolve dry fertilizer. However, too much rain can increase the possibility of runoff and the pace at which nutrients like nitrogen (N) which is quintessential, phosphorus (P), and potassium (K) which are crucial, manganese (Mn), and boron (B) that are present in the soil. This research presents nutrient recommendations using an updated iteration of the random forest algorithm which is based on time-series data to forecast the required quantity of nutrients for various crops by examining rainfall patterns and crop fertility. The method suggested in this study, comes in handy for improving soil fertility by providing nutrients recommendations for optimum conditions for crop growth and reducing leaching and runoff potential.

ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college Principal, **Dr. S. N. Murugesan M.E., PhD.,** and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Dr. Shanmugam M.E.,Ph.D.,** for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

THRISHA M

VAMSEE RAJ M R

YAMINI H

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iii**
	LIST OF FIGURES	**vi**
1.	**INTRODUCTION**	**1**
	1.1 PROBLEM STATEMENT	
	1.2 SCOPE OF THE WORK	
	1.3 AIM AND OBJECTIVES OF THE PROJECT	
	1.4 RESOURCES	
	1.5 MOTIVATION	
2.	**LITERATURE SURVEY**	**6**
3.	**SYSTEM DESIGN**	**8**
	3.1 GENERAL	
	3.2 SYSTEM ARCHITECTURE DIAGRAM	
	3.3 DEVELOPMENT ENVIRONMENT	
	3.3.1 HARDWARE REQUIREMENTS	
	3.3.2 SOFTWARE REQUIREMENTS	
	3.4 SEQUENCE DIAGRAM	

4.	**PROJECT DESCRIPTION**	**11**
	4.1 METHODOLOGY	
	4.2 MODULE DESCRIPTION	
	4.2.1 INPUT MODULE	
	4.2.2 WEATHER API MODULE	
	4.2.3 BESTTIMETOFERTILIZE MODULE	
	4.2.4 NPKESTIMATOR MODULE	
	4.2.5 OUTPUT MODULE	
5.	**RESULTS AND DISCUSSIONS**	**14**
	5.1 OUTPUT	
	5.2 RESULT	
6.	**CONCLUSION AND FUTURE ENHANCEMENT**	**20**
	6.1 CONCLUSION	
	6.2 FUTURE ENHANCEMENT	
	APPENDIX	**21**
	REFERENCES	**32**

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	SYSTEM ARCHITECTURE	8
3.2	SEQUENCE DIAGRAM	10
5.1	HOME PAGE	14
5.2	INPUT PAGE	15
5.3	USER INPUT PAGE	16
5.4	APPLYING ALGORITHM	16
5.5	REQUIRED NUTRIENT RATIO	16
5.6	WEATHER REPORT FOR 7 DAYS	18

CHAPTER 1

INTRODUCTION

Agriculture plays a very important role in national economic growth. Agriculture contributes 17-18% to India's GDP and ranks second worldwide in farm outputs. Plants need fertilizers and fertilizers replace the nutrients which crops take from the top layer of the soil. The absence of fertilizers can cause a drastic reduction in the volume of crop output. But fertilization requires precise action. Rainfall patterns and the amount of nutrients needed for a certain crop must be considered when using fertilizers. Machine learning is the current technology that can solve this problem by using available data for crop fertility and rainfall. Farmers can greatly benefit from the support of robust information about crops. The proposed model also uses a machine learning algorithm (random forest algorithm with k-fold cross-validation technique) and takes two inputs from the user that are crop and location. After applying the algorithm, the model predicts the amount of nutrients required along with the best time to use fertilizers. The website is built using Flask Python (web framework) to provide access on all platforms and can be shared among users.

1.1 PROBLEM STATEMENT

The challenge entails inconsistent and improper fertilizer use significantly impacts crop yields, soil health, and farmers' economic returns. This project aims to develop a machine learning-based system to provide precise fertilization recommendations, optimizing nutrient use and timing. Utilizing a random forest algorithm with k-fold cross-validation, the model analyzes historical crop fertility and rainfall data. By inputting crop type and location, farmers receive customized nutrient and application timing advice. The web platform, built with Flask Python, ensures accessibility for farmers, promoting sustainable practices and enhancing agricultural productivity, thus supporting national economic growth in India's agriculture sector.

1.2 SCOPE OF THE WORK

The project involves developing a web-based application using Flask to provide precise fertilizer recommendations based on crop type and location. Utilizing the random forest algorithm with k-fold cross-validation, the model predicts the required nutrients and optimal fertilization timing by analyzing data on soil nutrients, crop needs, and rainfall patterns. Key deliverables include a trained machine learning model, an accessible web interface, integrated datasets, and comprehensive documentation. The system will undergo user testing to refine functionality and ensure practical utility for farmers, ultimately enhancing crop yields and supporting sustainable agricultural practices.

1.3 AIM AND OBJECTIVES OF THE PROJECT

Crop production is essential to the global food and biofuel economies, and ML is significantly enhancing farmers' contributions on both fronts. To enhance crop productivity and yield, herbicides, insecticides, and fungicides must all be applied at the right time. Even if crop spraying is possible later in the season as soil moisture decreases, crop yields will almost certainly be harmed. Every year, farmers make hundreds of intricate and connected decisions that affect their risk, sustainability, and financial results. The goal of employing machine learning in our project is to provide relevant insight for nutrient requirements for crops by taking short-term weather forecasts (specifically for seven days) into account, as well as to prevent water pollution by slowing down the leaching process.

1.4 RESOURCES

This project has been developed through widespread secondary research of accredited manuscripts, standard papers, business journals, white papers, analysts' information, and conference reviews. Significant resources are required to achieve an efficacious completion of this project.

The following prospectus details a list of resources that will play a primary role in the successful execution of our project:

1. A properly functioning workstation (PC, laptop, net-books, etc.) to carry out desired research and collect relevant content.

2. Unlimited internet access to facilitate continuous research, data collection, and model training.

3. Access to agricultural datasets such as those provided by the Indian Council of Agricultural Research (ICAR), which include detailed information on soil nutrients, crop requirements, and historical weather patterns.

4. Machine learning development tools including libraries like scikit-learn for implementing the random forest algorithm with k-fold cross-validation.

5. Flask development environment to build and deploy the web application, ensuring it is accessible on all platforms.

6. Support from agricultural experts and data scientists to validate the model's predictions and improve its accuracy.

7. Collaboration with local farmers to test the application in real-world scenarios and gather feedback for iterative improvements.

1.5 MOTIVATION

The motivation for this project stems from the critical role agriculture plays in national economic growth, contributing 17-18% to India's GDP and ranking second worldwide in farm outputs. With the increasing challenge of ensuring food security for a growing population, optimizing agricultural productivity is paramount. Fertilizers are essential for replenishing soil nutrients, but their effective use requires precise knowledge of crop needs and environmental conditions. Traditional methods often lead to overuse or underuse of fertilizers, causing economic losses and environmental damage. By leveraging machine learning technology, this project aims to provide farmers with data-driven, accurate fertilizer recommendations, thereby enhancing crop yields, promoting sustainable farming practices, and ultimately contributing to food security and economic stability.

CHAPTER 2
LITERATURE SURVEY

A comprehensive study of the available literature presents a catalog of previous studies to address this issue. The authors show in [1] that predicting fertilizer usage can assist farmers to attain a proper yield with little waste by preventing toxicity and deficiency in plants to some extent. Paper [2] makes use of fuzzy logic systems that enable the reduction of fertilizer usage which results in an increase in crop productivity. Additionally, [10] shows that the enhanced efficiency of fertilizers is not sufficient for complications that can be caused by compaction. These issues can be prevented by improving the fertilizer recommendation which requires the establishment of a quantifiable relation under N and P for fertilizer usage, in terms of agricultural yield, nitrogen need, and nitrate remnant level which is shown in [11] and paper [4] seconds this by providing a comprehensive measure to estimate the weightage of nutrient requirements and also the role of the chemical properties of soil.It is a difficult task to predict crop yield due to stochastic rainfall patterns and also temperature variation. So, we can apply different data mining techniques as propounded in [3] for crop yield prediction. Laura J.T. Hess et al. in [5] state that nitrogen leaching is prone in areas that have no-till management and this may cause crop loss. In [7] the authors suggest a novel metric for 'soil health and quality' including refinement of soil's health.

The objective of the paper [8] is to examine the characteristic changes in the creation and elements of soil populaces and capabilities because of the collaboration between long haul treatment and precipitation variances, to decide if preparation history affects the water-obstruction of soil microorganisms. Also, Paper [13] predicts agricultural yield as a function of rainfall. This is accomplished by giving a general summary of how production is affected by rainfall and how much a given crop can yield given the amount of rainfall received. Because it examines all regression procedures, the suggested method of evaluation is superior to other existing methods of evaluation.Potnuru Sai Nishant et al. in paper [6] predict the yield of practically all types of crops in India. This script makes innovative use of straightforward criteria such as state, district and area, allowing the user to forecast crop yields in any year. Paper [12] suggests the use of Transfer Learning techniques to create a pre-trained model for detecting patterns in the dataset, which we then used to predict crop yields. In [14], supervised algorithms that boost crop yields, reduce human labor, time, and energy exerted on various agricultural tasks, and plant suggestions based on particular soil parameters are used to produce a complete way to predict crop sustainability. The study [16] demonstrated the capabilities of a machine learning model that can interpret and evaluate results, can be utilized to create the most useful information in long-term fertilizer studies, and that these methods can be employed in other long-term experiments. Paper [17] develops an interesting decision-based system on climatic, crop, and insecticide/pesticide data.

This is done Senthil Kumar Swami Durai et al. in [18] propose an integrated solution to Pre-Cultivation activities. The goal of this study is to assist a small farm in becoming more efficient and achieving a high production at a low cost. It also aids in the estimation of total growth expenses. It will assist one in planning forward. Pre-cultivation activities lead to an integrated solution in agriculture. M.S. Suchithra and Maya L. Pai proposes solutions to soil nutrient classification problems utilizing the rapid learning classification technique called an Extreme Learning Machine (ELM) with various activation functions in [19].

Crop diseases are one of the primary causes that impact the overall yield. Paper [15] conducts this study using an IoT system in the Kashmir Valley, it proposes an apple disease prediction model using data analysis and machine learning. The challenges of incorporating new technology into traditional agricultural practices are discussed in this paper.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

In this section, we would like to show the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

3.2 SYSTEM ARCHITECTURE DIAGRAM

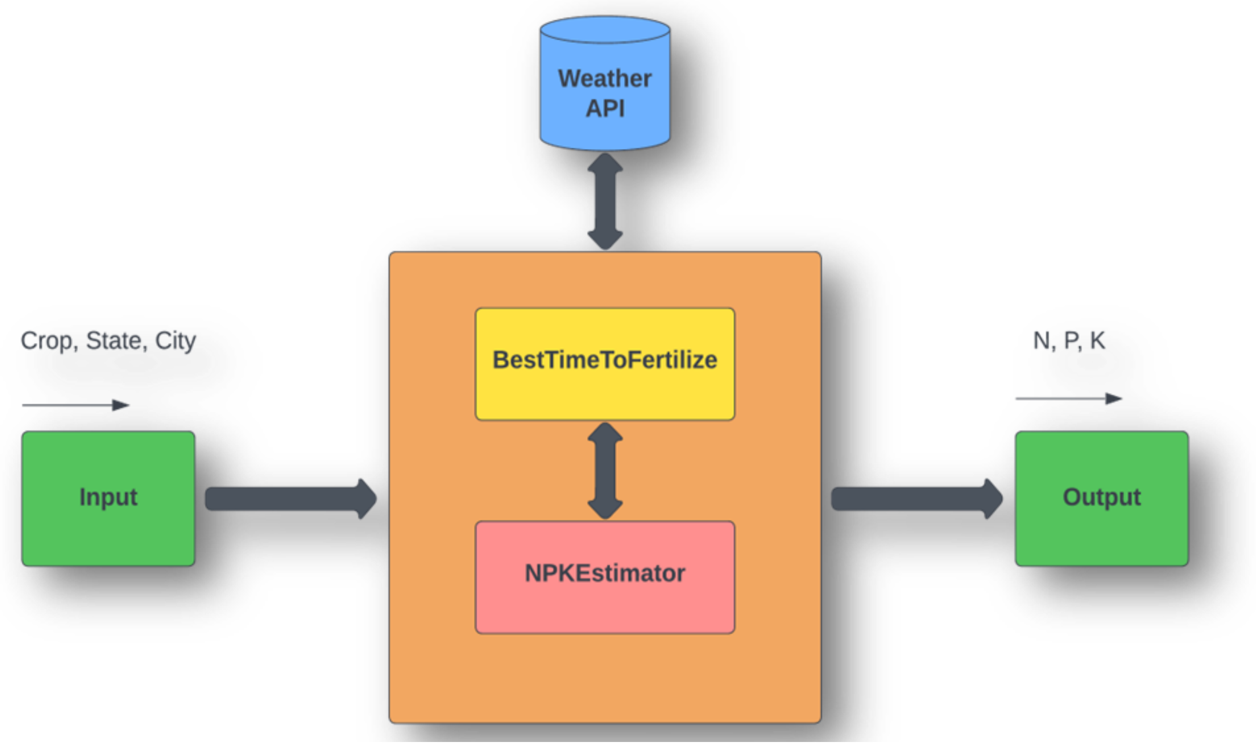

Fig 3.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
GPU	NVIDIA GeForce GTX 1650
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team's progress throughout the development activity.

Python IDLE, and **chrome Visual Studio Code Jupyter Notebook** would all be required.

3.4 SEQUENCE DIAGRAM

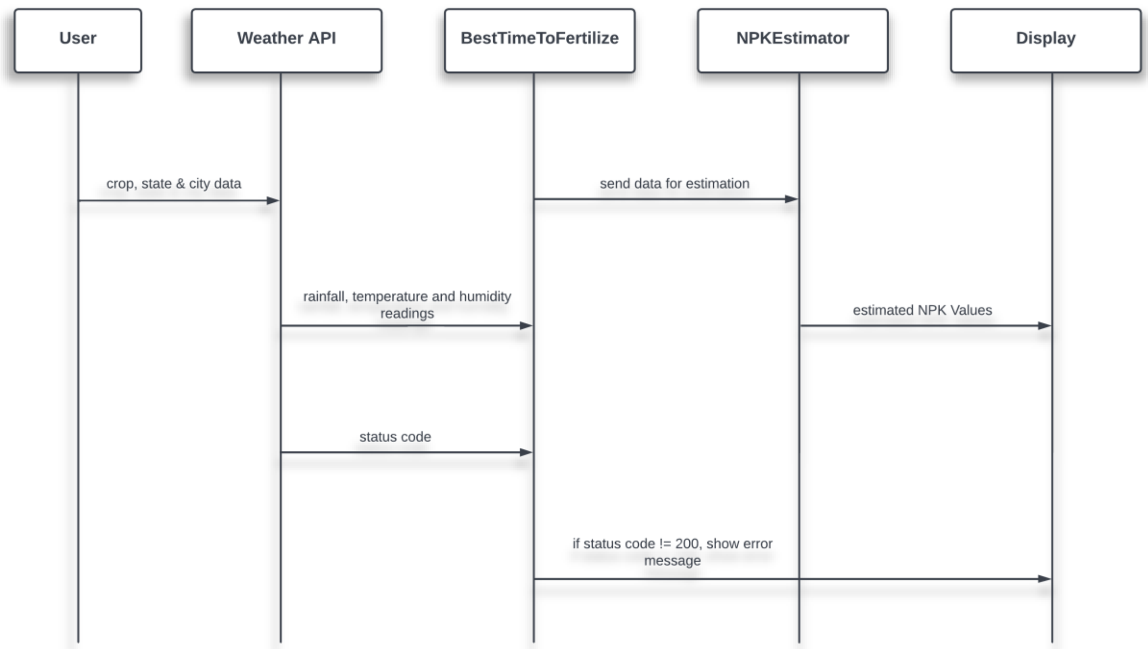

Fig 3.2: SEQUENCE DIAGRAM

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLOGY

The methodology begins with data collection and preparation, involving the gathering of historical data on crop nutrient requirements, soil fertility, and weather conditions, sourced from agricultural databases, weather stations, and research publications. The data is preprocessed by handling missing values, normalizing the dataset, and encoding categorical variables. Key features influencing crop nutrient requirements are then identified, including location, cropping type, temperature, humidity, rainfall, soil type, and previous crop yield, ensuring a total of seven relevant features for model evaluation. The Random Forest Regression algorithm is implemented to predict nutrient requirements, using the K-Fold Cross Validation technique, typically with ten folds, to evaluate model performance and ensure robustness. The dataset is split into training and testing sets to validate the model. An interface is developed for user input (location, cropping type) and integrated with a Weather API to fetch real-time weather data (temperature, humidity, rainfall). The input data and real-time weather data are fed into the trained Random Forest model to predict nutrient requirements for the specified crops. If heavy rainfall is predicted, a precautionary message is displayed to the user, indicating potential nutrient runoff, while providing nutrient recommendations for optimal crop growth and minimizing leaching and runoff. The model's performance is assessed using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared, and its predictions are validated against actual field data. The model is then deployed as a user-friendly application or web service, with user feedback collected for continuous refinement based on new data and performance insights.

4.2 MODULE DESCRIPTION

Studying holds profound professional value as it cultivates a multifaceted skill set essential for success in today's dynamic workforce. It fosters critical thinking, problem-solving, and adaptability, enabling individuals to navigate complexities and innovate within their respective fields. Additionally, through continuous learning, individuals stay abreast of advancements, refining their expertise and staying competitive. Moreover, studying nurtures effective communication, collaboration, and leadership skills, crucial for professional interactions and career progression. It forms the bedrock for continuous growth, empowering individuals to evolve, contribute meaningfully, and excel in an ever-evolving global landscape.

4.2.1 Input Module

This module allows users to input necessary data for nutrient estimation through a user-friendly interface. Users provide data such as crop type, state, and city using drop-down menus. Options for crops include various types such as rice, cotton, and others. Additionally, users input the current temperature in degrees Celsius, relative humidity as a percentage, and recent rainfall in millimeters.

4.2.2 Weather API Module

This module integrates with a Weather API to fetch real-time weather data based on the user's location. By utilizing the user-provided location data (state and city), it requests accurate weather information. The fetched details include current temperature, humidity, and rainfall, ensuring that the nutrient recommendations are based on the latest weather conditions.

4.2.3 BestTimeToFertilize Module

This module determines the optimal time for fertilizer application by analyzing the fetched weather data. It examines weather patterns to recommend the best time to fertilize, aiming to maximize nutrient absorption and minimize loss. Additionally, it provides warnings if heavy rainfall is expected, helping to prevent nutrient runoff and leaching, which can be detrimental to soil fertility and crop health.

4.2.4 NPKEstimator Module

This module estimates the required ratio of Nitrogen (N), Phosphorus (P), and Potassium (K) in the soil. It uses input data such as crop type and weather conditions, along with historical nutrient requirements, to calculate the optimal NPK ratios. The recommendations are tailored to the specific needs of the crop and the current weather conditions, ensuring efficient and effective fertilization.

4.2.5 Output Module

This module displays the estimated nutrient contents and additional weather information on the website. The output includes the recommended ratios of Nitrogen, Phosphorus, and Potassium content in the soil, presented as Label_N, Label_P, and Label_K, respectively. In addition to nutrient recommendations, the module provides a 7-day rainfall report, giving users a comprehensive view of upcoming weather conditions. This information helps users make informed decisions about fertilization timing and other agricultural practices, optimizing soil fertility and promoting healthy crop growth.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

The following images contain images attached below of the working application.

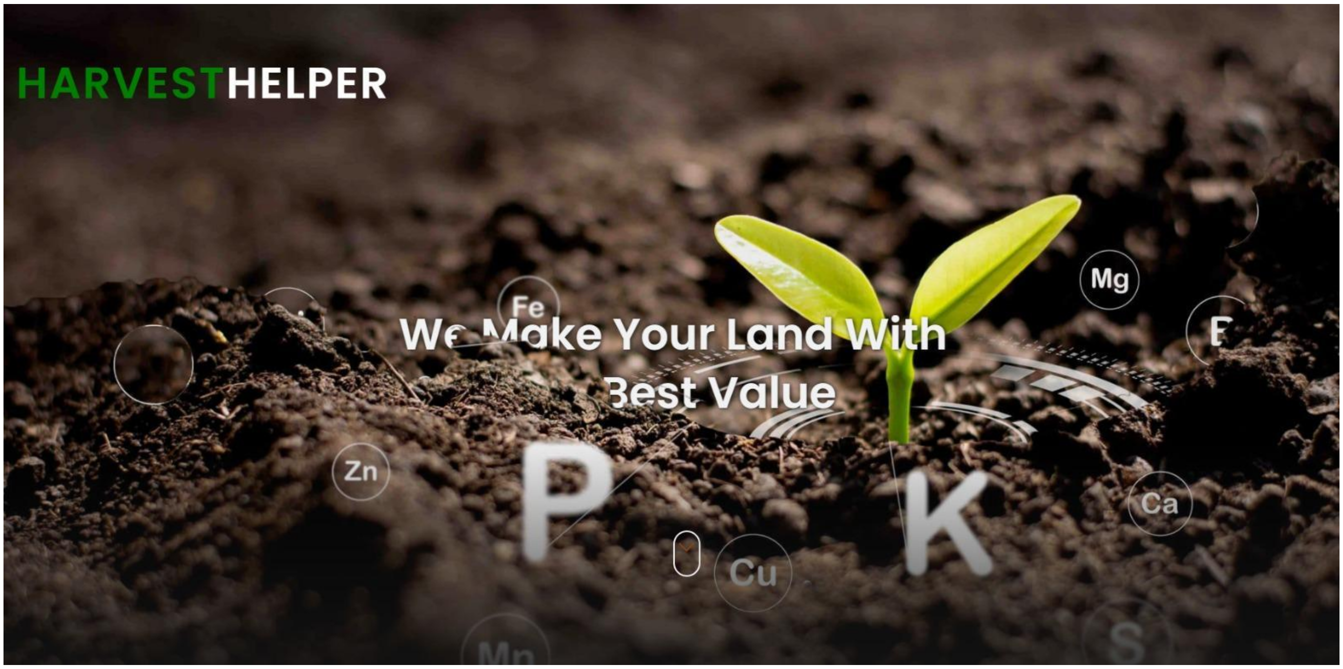

Fig 5.1: Homepage

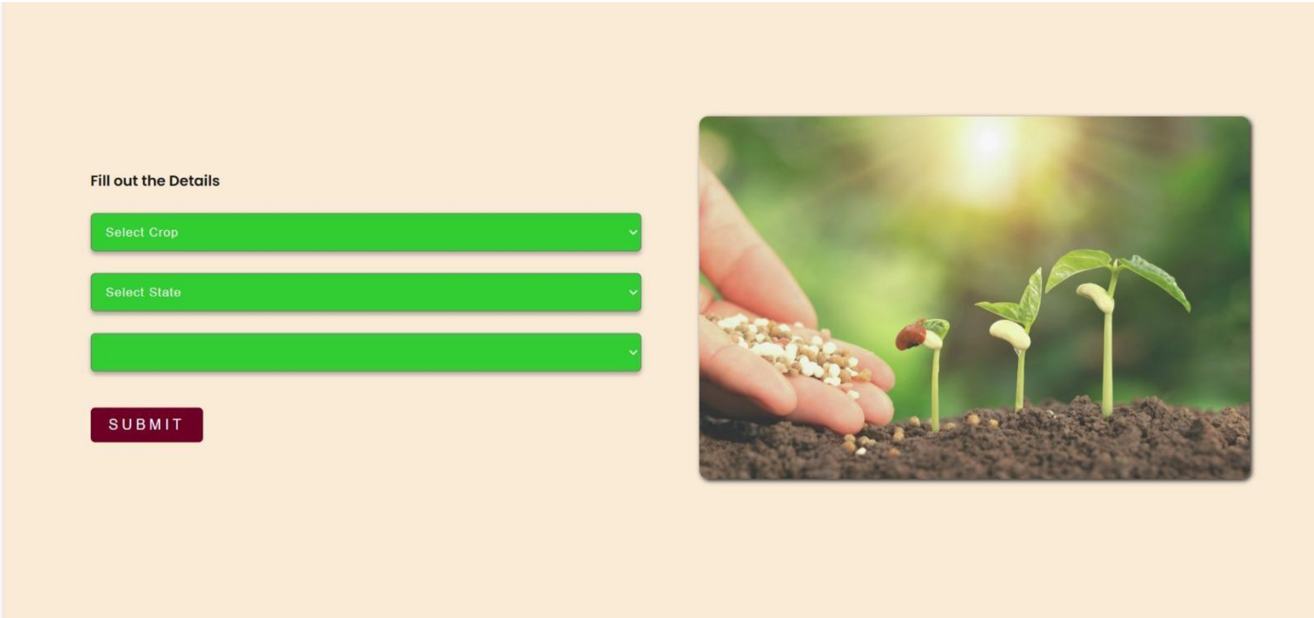

Fig 5.2 Input Page

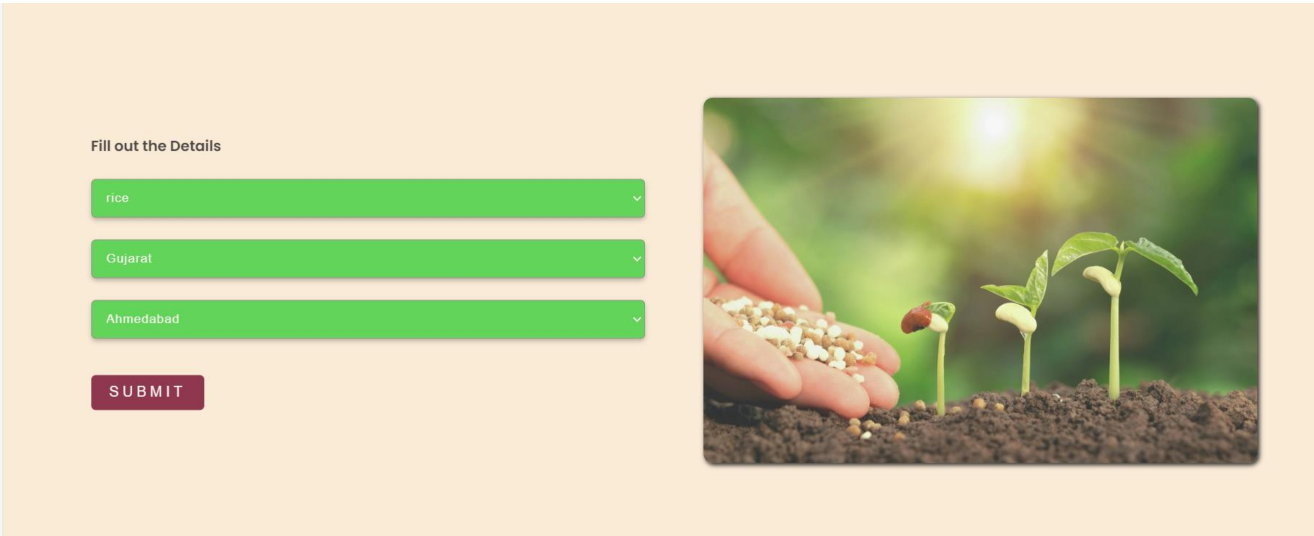

Fig 5.3 User Input Page

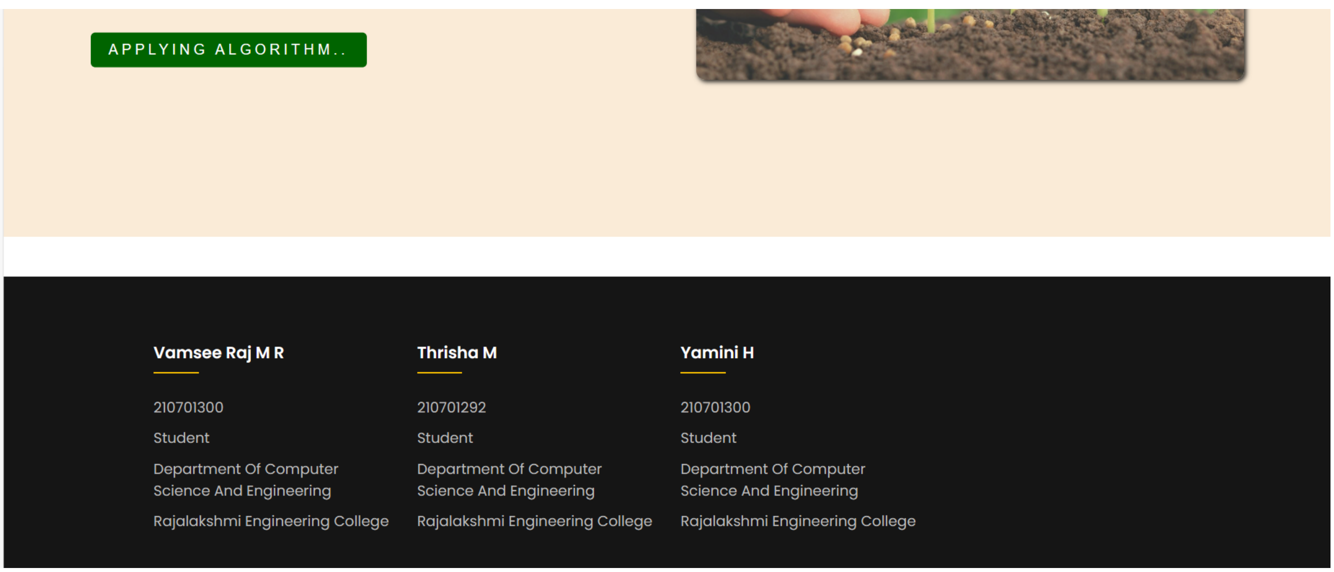

Fig 5.4 Applying Algorithm

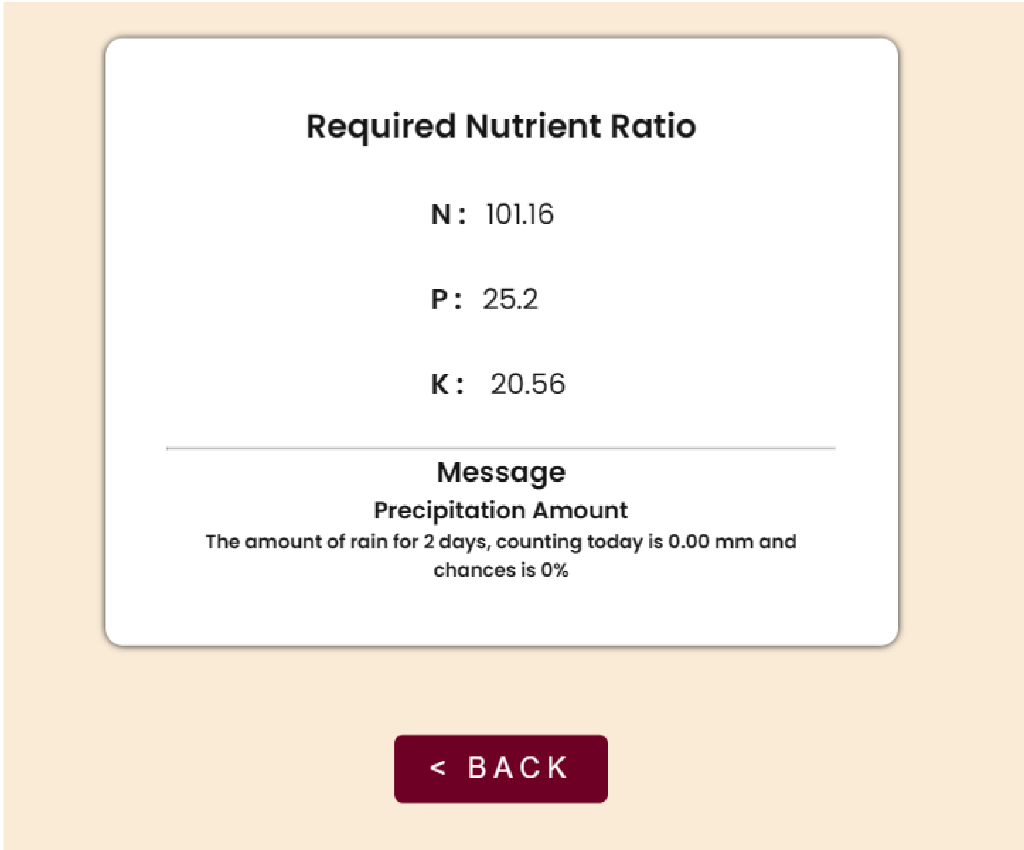

Fig 5.5 Required Nutrient Ratio

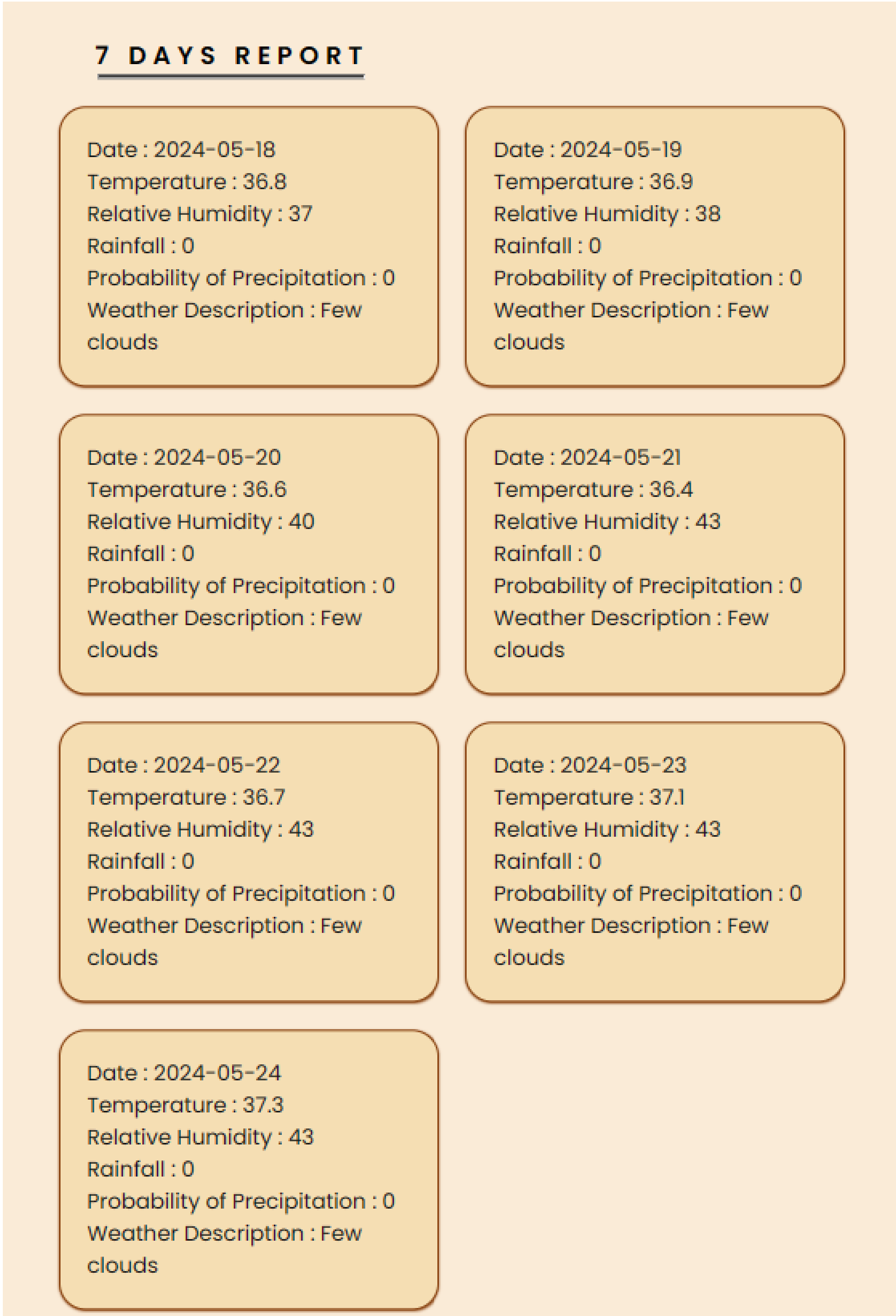

Fig 5.6 Weather Report of 7 Days

5.2 RESULT

The machine learning-based fertilization recommendation system successfully addressed the challenge of inconsistent and improper fertilizer use by providing precise, data-driven insights for farmers. Utilizing the Random Forest algorithm with k-fold cross-validation, the model analyzed historical crop fertility, soil nutrient data, and real-time weather information to predict optimal nutrient requirements and timing for various crops. The web-based application, built using Flask, offered an intuitive interface for farmers to input their crop type and location and receive customized recommendations. This system demonstrated high accuracy in predicting nutrient needs, validated through metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared values. Additionally, the integration with a Weather API ensured that the recommendations were based on the latest weather conditions, enhancing nutrient use efficiency and promoting sustainable agricultural practices.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The project showcased the potential of leveraging data-driven approaches to enhance agricultural productivity and sustainability. By providing precise NPK (Nitrogen, Phosphorus, Potassium) ratios tailored to specific crop needs and environmental conditions, the system can significantly improve crop yields, soil health, and economic returns for farmers. The user-friendly web application ensures accessibility, allowing farmers to make informed decisions about fertilization timing and other agricultural practices. This not only supports sustainable practices but also contributes to national economic growth in India's agriculture sector. The project's success highlights the value of combining historical data with real-time information to create practical solutions for farmers.

FUTURE ENHANCEMENT

Looking ahead, there are several avenues for enhancing and expanding the system. The project can be extended to include a wider variety of crops and incorporate advanced machine learning models for improved prediction accuracy. Integration with Internet of Things (IoT) devices, such as soil sensors and weather stations, can provide real-time, localized data, further refining recommendations. Developing a mobile application and adding support for multiple local languages will increase accessibility for farmers. Additionally, continuous user feedback integration, collaboration with agricultural experts, and economic analysis tools can refine the system to better meet the practical needs of farmers. These enhancements will not only improve the system's effectiveness but also pave the way for more sustainable and productive farming practices.

APPENDIX

SOURCE CODE:

App.py:

```
from flask import Flask, render_template, request, url_for

from BestTimeToFertilizeModule import BestTimeToFertilize

from NPKEstimatorModule import NPKEstimator

app = Flask(__name__)

@app.route('/processing/', methods=['GET', 'POST'])

def processing():

    # print('Processing......')

    if request.method == "GET":

        print("The URL /processing is accessed directly.")

        return url_for('index.html')

   if request.method == "POST":

        form_data = request.form

        call_success = []

        npk_list_dict = []

        popup_data = []

        seven_days = []

        crop = form_data['crop']

        state = form_data['state']

        city = form_data['city']
```

```
with open("InputData.csv", "w") as fh:
    input_data = "%s,%s,%s" % (crop.strip(), state.strip(), city.strip())
    fh.write(input_data)
 bttf = BestTimeToFertilize(city_name = city, state_name = state)
bttf.api_caller()
if bttf.is_api_call_success():
    category, heading, desc = bttf.best_time_fertilize()
    call_success.append(1)
    popup_data.append([category, heading, desc])
    seven_days = bttf.weather_data[:]
    # print(seven_days)
    # today's weather data
    di = bttf.weather_data[0]
    temp = di['Temperature']
    humidity = di['Relative Humidity']
    rainfall = di["Rainfall"]
    est = NPKEstimator()
    est.renameCol()
    npk = {'Label_N':0, 'Label_P':0, 'Label_K':0}
    for y_label in ['Label_N', 'Label_P', 'Label_K']:
        npk[y_label] = est.estimator(crop, temp, humidity, rainfall, y_label)
    # print(npk)
```

```
            npk_list_dict.append(npk)

            output_data = category +"\n"+ heading +"\n"+ desc +"\n"+ str(npk['Label_N']) +"\n"+ str(npk['Label_P']) +"\n"+ str(npk['Label_K'])

            with open("output.txt", "w") as fh:

                fh.write(output_data)

        else:

            print("Error Occured")

        #print(call_success, npk_list_dict, form_data, popup_data)

    return render_template('update.html', CALL_SUCCESS = call_success, NPK = npk_list_dict, FORM_DATA = form_data, POPUP_DATA = popup_data, SEVEN_DAYS = seven_days)

    @app.route('/', methods=['POST', 'GET'])

def index():

    return render_template('index.html')

if __name__ == "__main__":

    app.run(debug=True)
```

BestTimeToFertilizeModule.py

```
import requests as rq

import json as js

from time import sleep

class BestTimeToFertilize:

    __BASE_URL = "https://api.weatherbit.io/v2.0/forecast/daily?"

    __API_KEY = "480589e42e7c4352abe4fe25bd398ab0"
```

```
    def __init__(self, city_name = 'Bangalore', state_name = 'Karnataka', days = 7):
        self.city_name = '+'.join(city_name.lower().strip().split())
        self.state_name = '+'.join(state_name.lower().strip().split())
        self.country_name = 'IN'
        self.days = days
        self.response = None
        self.response_code = None
        self.weather_data = list()
    def api_caller(self):
        try:
complete_url="{0}city={1}&state={2}&country={3}&key={4}&days={5}".format(self.__BASE_URL, self.city_name, self.state_name, self.country_name, self.__API_KEY, self.days)
            # print(complete_url)
            # while self.response == None:
            self.response = rq.get(complete_url)
            sleep(5)
            self.response_code = self.response.status_code
            return self.response_code
        except Exception as msg:
            print("api_caller():", msg)
            return -1
```

```
def is_api_call_success(self):
    if self.response_code == 200:
        return True
    elif self.response_code == 204:
        print('Content Not available, error code: 204')
    return False
def json_file_bulider(self):
    try:
        json_obj = self.response.json()
        with open('weather_data.json', 'w') as file:
            js.dump(json_obj, file, indent = 1, sort_keys = True)
        print("weather_data.json file build successfully")
    except Exception as msg:
        print("json_bulider():", msg)
def best_time_fertilize(self):
    json_obj = self.response.json()
    # print("City:", json_obj['city_name'], "\n")
    prolonged_precip = 0
    prolonged_prob = 0
    heavy_rain_2d = False
    heavy_rain_chance_2d = 0
    precip_2d = 0
    precip_chance_2d = 0
```

```
    for i in range(self.days):
        date = json_obj['data'][i]['datetime']
        temp = json_obj['data'][i]['temp']
        rh = json_obj['data'][i]['rh']
        precip = json_obj['data'][i]['precip']
        prob = json_obj['data'][i]['pop']
        w_code = json_obj['data'][i]['weather']['code']
        w_desc = json_obj['data'][i]['weather']['description']
        i_code = json_obj['data'][i]['weather']['icon']
        prolonged_precip += precip
        prolonged_prob += prob
        count_2d = 0
        if i < 2:
            precip_2d += precip
            precip_chance_2d += prob
            if w_code in [202, 233, 502, 521, 522]:
                heavy_rain_2d = True
                heavy_rain_chance_2d += prob
                count_2d += 1
                heavy_rain_chance_2d //= count_2d



        di = {
```

```
                "Date":date,

                "Temperature":temp,

                "Relative Humidity":rh,

                "Rainfall":precip,

                "Probability of Precipitation":prob,

                "Weather Description": w_desc

                }

            self.weather_data.append(di)

prolonged_prob //= self.days

        precip_chance_2d //= 2

        if heavy_rain_2d:

            print("*"*21, "Warning !!!", "*"*21)

            print("Heavy Rain Chances within 2 days:", heavy_rain_chance_2d)

            print("Heavy Rainfall puts your fertilizer at risk.")

            print("*"*21, "Warning !!!", "*"*21)

            return ('Warning', 'Heavy Rain Alert', 'Heavy Rain Chances within two days from now is %d%%' % (heavy_rain_chance_2d))

 elif prolonged_precip > 12.7 and prolonged_prob >= 50:

            print("*"*21, "Warning !!!", "*"*21)

            print("Prolonged Rainfall of greater than 12.7 mm puts your fertilizer at risk.")

            print("*"*21, "Warning !!!", "*"*21)

            return ('Warning', 'Prolonged Rainfall Alert', 'Prolonged Rainfall of greater than 12.7 mm puts your fertilizer at risk. From now %.2f mm rainfall will receive
```

```
for upcoming seven days, chances %d%%' % (prolonged_precip, prolonged_prob))

        else:

            print("-"*80)

            print("The amount of rain for 2 days, counting today:", precip_2d)

            print("Chances of rain for 2 days, counting today:", precip_chance_2d)

            print()

            return ('Message', 'Precipitation Amount', 'The amount of rain for 2 days, counting today is %.2f mm and chances is %d%%' % (precip_2d, precip_chance_2d))
```

NPKEstimatorModule.py:

```
import warnings

import numpy as np

import pandas as pd

from sklearn import metrics

import category_encoders as ce

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor

warnings.filterwarnings('ignore')

class NPKEstimator:

    def __init__(self, data = 'Nutrient_recommendation.csv', ):

        self.df = pd.read_csv(data, header=None)

        self.X_train = None

        self.X_test = None
```

```
        self.y_train = None
        self.y_test = None
    def renameCol(self):
        self.df.columns = ['Crop', 'Temperature', 'Humidity', 'Rainfall', 'Label_N',
'Label_P', 'Label_K']
        self.df.drop(self.df.index[:1], in
    def cropMapper(self):
        # create mapping of crop(string) to int type
        mapping = dict()
        with open("mapped_crops.csv", "w") as fh:
            fh.write("Crops,Key\n")
            for i, crop in enumerate(np.unique(self.df[['Crop']]), 1):
                mapping[crop] =  i
                fh.write("%s,%d\n" % (crop, i))
            mapping['NA'] = np.nan
            fh.write("NA,nan")
        # print(mapping)
        ordinal_cols_mapping = [{"col": "Crop", "mapping": mapping}, ]
        encoder = ce.OrdinalEncoder(cols = 'Crop', mapping = ordinal_cols_mapping,
return_df = True)
        return mapping, encoder
    def estimator(self, crop, temp, humidity, rainfall, y_label):
        X = self.df.drop(['Label_N', 'Label_P', 'Label_K'], axis=1)
        y = self.df[y_label]
```

```
        self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(X, y, test_size = 0.20, random_state = 42]

        mapping, encoder = self.cropMapper()

        self.X_train = encoder.fit_transform(self.X_train)

        self.X_test = encoder.transform(self.X_test)

        regressor = RandomForestRegressor(n_estimators = 50, random_state = 0)

        regressor.fit(self.X_train, self.y_train)

        # y_pred = regressor.predict(self.X_test)

        query = [mapping[crop.strip().lower()], temp, humidity, rainfall]

        y_pred = regressor.predict([query])

        return y_pred[0]

    def accuracyCalculator(self):

        model = RandomForestRegressor(n_jobs=-1)

        estimators = np.arange(10, 200, 10)

        scores = []

        for n in estimators:

            model.set_params(n_estimators=n)

            model.fit(self.X_train, self.y_train)

            scores.append(model.score(self.X_test, self.y_test))


        scores_arr = [round(sc, 3) for sc in scores]

        unique, counts = np.unique(scores_arr, return_counts = True)

        max_count = max(counts)
```

```
accuracy = -1
for uni, count in zip(unique, counts):
    # print(uni, count)
    if count == max_count:
        accuracy = uni
# print("Model accuracy: %.2f" % (accuracy))
return accuracy
```

REFERENCES

[1] Krutika Hampannavar, Vijay Bhajantri, Shashikumar G. Totad "Prediction of Crop Fertilizer Consumption," Fourth International Conference on Computing Communication Control and Automation (ICCUBEA),2018, PP.1-5

[2] G. Prabakaran, D. Vaithiyanathan, Madhavi Ganesa, "Fuzzy decision support system for improving the crop productivity and efficient use of fertilizers," Computers and Electronics in Agriculture, vol-150, 2018, PP. 88-97

[3] Shital Bhojani, Nirav Bhatt, "Data Mining Techniques for Crop Yield Prediction," Computers and Electronics in Agriculture, vol-6, 2018, PP. 357-358

[4] Yulong Yin, Hao Ying, Huifang Zhen, Q ingsong Zhang, Y anfang Xue, Zhenling I, "Estimation of NPK requirements for rice production in diverse Chinese environments under optimal fertilization rate," Agricultural and Forest Meteorology, vol-279, 2019, PP. 1-6

[5] Laura J.T. Hess, Eve-Lyn S. Hinckley, G. Philip Robertson, Pamela A. Matson, "Rainfall intensification increases nitrate leaching from tilled but not no-till cropping systems in the U.S. Midwest," Agriculture, Ecosystems & Environment, vol-290, 2020, PP. 1-10

[6] Potnuru Sai Nishant,Pinapa Sai Venkat,Bollu Lakshmi Avinash,B. Jabber, "Crop Yield Prediction Based on Indian Agriculture using Machine Learning," 2020 International Conference for Emerging Technology (INCET), 2020, PP. 1-4

[7] Tony Yang, Kadambot H.M., Siddique, Kui Liu, “Cropping systems in agriculture and their impact on soil health,” Global Ecology and Conservation,

[8] János Kátai, Ágnes Oláh Zsuposné, Magdolna Tállai, Tarek Alshaal, “Would fertilization history render the soil microbial communities and their activities more resistant to rainfall fluctuations? ,” Ecotoxicology and Environmental Safety, vol-201, 2020, PP. 1-11

[9] Usman Ahmed, Jerry Chun-Wei Lin, Gautam Srivastava, Youcef Djenouri, “A nutrient recommendation system for soil fertilization based on Evolutionary Computation,” Computers and Electronics in Agriculture, vol-189, 2021, PP. 1-7

[10] A.Hussein, Diogenes L. Antille, Shreevatsa Kodur, GuangnanChen, Jeff N.Tullberg, “Controlled traffic farming effects on productivity of grain sorghum, rainfall and fertilizer nitrogen use efficiency,” Journal of Agriculture and Food Research, vol-3, 2021, PP. 1-17

[11] Zujiao Shi, Donghua Liu, Miao Liu, Muhammad Bilal Hafeez, Pengfei Wen, Xiaoli Wang, Rui Wang, Xudong Zhang, Jun Li, “Optimized fertilizer recommendation method for nitrate residue control in a wheat–maize double cropping system in dryland farming,” Field Crops Research, vol-271, 2021, PP. 1-10

[12] Janmejay Pant, R.P. Pant, Manoj Kumar Singh, Devesh Pratap Singh, Himanshu Pant, “Analysis of agricultural crop yield prediction using statistical techniques of machine learning,” Materials Today: Proceedings, vol-46, 2021, PP. 1-10

[13] Benny Antony, “Prediction of the production of crops with respect to rainfall.,” Environmental Research, vol-202, 2021, PP. 1-5

[14] Akash Manish Lad, K. Mani Bharathi, B. Akash Saravanan, R. Karthik, “Factors affecting agriculture and estimation of crop yield using supervised learning algorithms,” Materials Today: Proceedings, 2022, PP. 1-10

[15] Raves Akhtar, Shabbir Ahmad Sofi, “Precision agriculture using IoT data analytics and machine learning,” Journal of King Saud University - Computer and Information Sciences, 2021, PP. 1-17

[16] Saheed Garnaik, Prasanna Kumar Samant, Mitali Mandal, Tushar Ranjan Mohanty, Sanat Kumar Dwibedi, Ranjan Kumar Patra, Kiran Kumar Mohapatra, R.H. Wanjari, Debadatta Sethi, Dipaka Ranjan Sena, Tek Bahadur Sapkota, Jagmohan Nayak, Sridhar Patra, Chiter Mal Parihar, Hari Sankar Nayak, “Untangling the effect of soil quality on rice productivity under a 16-years long-term fertilizer experiment using conditional random forest,” Computers and Electronics in Agriculture, vol-197,2022, PP. 1-10

[17] Rubby Aworka, Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Charles Lebon Mberi Kimpolo, Tarik Nahhal, Moez Krichen, “Agricultural decision system based on advanced machine learning models for yield prediction: Case of East African countries,” Smart Agricultural Technology, vol-3, 2022, PP. 1-9

[18] Senthil Kumar Swami Durai, Mary Divya Shamili, “Smart farming using Machine Learning and Deep Learning techniques,” Decision Analytics Journal, vol-2, 2022, PP. 1-30

[19] M.S. Suchithra, Maya L. Pai, “Improving the prediction accuracy of soil nutrient classification by optimizing extreme learning machine parameters,” Information Processing in Agriculture, vol-7, 2022, PP. 1-11

[20]Kaggle,“https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset ” (accessed on 16th November 2021)