"Crop Recommendation System"

A Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of
Bachelor of Technology
(*Computer Science and Engineering*)

Submitted By:
Ansh Joshi(0827CS201036)
Anurag Mahajan(0827CS201037)
Anushka Bhanpiya(0827CS201040)
Avani Jain(0827CS201050)

Guided By:
Prof. Preeti Shukla
Department of computer Science And Engineering
AITR, Indore

Acropolis Institute of Technology & Research, Indore
January - June 2023

Examiner Approval

The Project entitled *"**Crop Recommendation System**"* submitted by **Ansh Joshi(0827CS201036), Anurag Mahajan(0827CS201037), Anushka Bhanpiya(0827CS201040), , Avani Jain(0827CS201050)** has been examined and is hereby approved towards partial fulfillment for the award of *Bachelor of Technology degree in Computer Science and Engineering* discipline, for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion is drawn therein, but approves the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date**

Recommendation

This is to certify that the work embodied in this minor project entitled **"*CROP RECOMMENDATION SYSTEM*"** submitted by **Ansh Joshi(0827CS201036), Anurag Mahajan(0827CS201037), Anushka Bhanpiya(0827CS201040), Avani Jain(0827CS201050)** is a satisfactory account of the bonafide work done under the supervision of ***Prof. Preeti Shukla***, is recommended towards partial fulfillment for the award of the Bachelor of Technology (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

(Dean Academics)

Students Undertaking

This is to certify that a project entitled "CROP RECOMMENDATION SYSTEM" has been developed by us under the supervision of **PROF. PREETI SHUKLA.** The whole responsibility of the work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University /Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

ANSH JOSHI(0827CS201036)

ANURAG MAHAJAN(0827CS201037)

ANUSHKA BHANPIYA(0827CS201040)

AVANI JAIN(0827CS201050)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach shore safely.

There are a number of people without whom this project's workwould not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, a deep sense of reverence, and respect to our guide and mentor PROF. PREETI SHUKLA, Professor, AITR, Indore for her motivation, sagacious guidance, constant encouragement, vigilant supervision, and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to PROF. KAMAL KUMAR SETHI, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful toother faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from Dr S C Sharma, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

Executive Summary

CROP RECOMMENDATION SYSTEM

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of the Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of *PROF. PREETI SHUKLA*.

Key Words: Image Processing, Analysis, Machine Learning, Learning Algorithm, Crop Prediction**.**

“Where the vision is one year, cultivate flowers;

Where the vision is ten years, cultivate trees;

Where the vision is eternity,

cultivate people.”
- Oriental Saying

LIST OF FIGURES

Table of Contents

Chapter 1 . Introduction

This chapter gives a description of machine learning, tasks involved, models, advantages, disadvantages, the objective of the proposed work, and a thesis overview.

India is the leading producer of a few crops. The predominant occupation in India is agriculture. The soil quality of agriculture includes the soil properties related to organic matter such as N(Nitrogen), C(Carbon), Ph(Phosphorus), Mg(Magnesium), Ca(Calcium), and K(Potassium).

The main aim is to analyze and predict the crop using soil properties parameters. In this application, the farmer provides the soil composition i.e., properties of the soil then the supervised machine learning algorithm will analyze and predict the right information as what crops has to be grown with their appropriate fertilizer details.

1.1 Overview

Agriculture planning plays a significant role in the economic growth and food security of the agro-based country. And still agriculture is being plagued by various problems. Farming practices and other activities of agriculture consume time, money as well as the efforts of a farmer.

1.2 Background and Motivation

Farmers face alot of issues with respect to the selection of the crop to be grown. If the wrong crop is grown thenfarmers have to deal with financial crisis. To feed the world population, an essential developmentin the agriculture production is important. Selection of crop is an important issue for agriculture planning. To overcome this problem, many researchers studied prediction of crop, soil classification and crop classification for agriculture planning using statistics methods or data mining techniques. But prediction of the appropriate crop for a soil is still a puzzle. There are many emerging techniques; the most prominent one is the machine learning technique. In this proposed work, the aim is to predict the right crop and appropriate fertilizer by analyzing the composition and other properties of the soil. For this machine learning algorithm is used, which will overcome the existing problems and increase the accuracy rate. Crop Prediction helps farmers in selecting proper crop for plantation and thus maximizing the yield rate and their earnings.

1.3 Problem Statement and Objectives

An application to predict crops according to rainfall, humidity, and soil type. It also predicts fertilizers needed.

1.4 Scope of the Project

The desired results from the proposed work can be met only when the right soilcomposition is provided by the farmer. The constraint in the proposed work is that the farmer has to externally get the soil sample for the soil composition tested and provide the test results to get the details about the right crop to be predicted.

1.5 Team Organization

ANURAG MAHAJAN: I investigated and found the right technology and studied in deep about it. For the implementation of the project, I also collected the object data. Front design is also a part of the work done by me in this project.

ANUSHKA BHANPIYA: Implementation logic for the project objective and coding of internal functionalities is done by me. Also, worked on Back end design .

ANSH JOSHI: I also worked on the implementation of React framework and integration of that with all the machine learning models involved in our underlying project.

AVANI JAIN: Along with doing a preliminary investigation an understanding the limitations of the current system, I studiedabout the topic and its scope and surveyed various research papers. I have also created a model for the same.

1.6 Report Structure

The project *Crop Prediction System* is primarily concerned with crop recommendation and the whole project report is categorized into four chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of the project which is then subsequently ended with report outline.

Chapter 2*:* Literature Survey- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different the Project. It also includes different Technology/ Techniques/ Tools/ Programming Languages used in developing with along project. The chapter ends with evaluation of project on different parameters like accuracy andefficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement**.**

Chapter 2. Literature Review

2.1 An Efficient Analysis of Crop Yield Prediction

Authors: Shriya Sahu, Meenu Chawla and Nilay Khare

Title: An Efficient Analysis of Crop Yield Prediction using Hadoop Framework Based on Random Forest Approach

Published in: IEEE (2017)

Methodology: In this paper, various parameters are considered from soil toatmosphere for predicting the suitable crop. Soil parameters such as type, ph level, iron, copper, manganese, sulphur, organic carbon, potassium, phosphate, nitrogen are considered. The random forest algorithm is used to classify the dataset which provides result in good accuracy with poor error rate. Since this framework can handle large dataset by processing it in MapReduce programming model. The phases of the proposed work are: Data Collection, Data Classification(Random Forest Algorithm), Hadoop Framework – MapReduce programming model and Final Prediction. The implementation is carried out in ubuntu 14.04 LTS with Hadoop 2.6.0 and the dataset is collected from various online sources to predict the suitable crop.

Accuracy: The accuracy achieved by this methodology is 91.43%.

Future Work: The future research will be devoted to predicting the required fertilizer and pesticide ratio based on the atmospheric and soil parameters for the farm land.

2.2 Crop Selection Method to Maximize Crop yield rate

Authors: Rakesh Kumar, M.P. Singh, Prabhat Kumar and J.P. Singh**.**

Title: Crop Selection Method to Maximize Crop yield rate using Machine Learning Technique

Published in: IEEE (2015)

Methodology: This work presents a technique named CSM to select sequence ofcrops to be planted over season. CSM method may improve net yield rate of crops to be planted over season. The proposed method resolves selection of crop (s) based on prediction yield rate influenced by parameters (e.g. weather, soil type, water density, crop type). The crop sowing table data considered are gathered from farmer of Patna District, Bihar (India). It takes crop, their sowing time, plantation days and predicted yield rate for the season as input and finds a sequence of crops whose production perday are maximum over season.

Accuracy: Performance and accuracy of CSM method depends on predicted value of influenced parameters.

Future Work: There is need to adopt a prediction method with more accuracy and high performance.

2.3 Analysis of Soil Behaviour and Prediction of Crop Yield

Authors: Monali Paul, Santosh K. Vishwakarma and Ashok Verma

Title: Analysis of Soil Behaviour and Prediction of Crop Yield using Data Mining Approach

Published in: IEEE (2015)

Methodology: In this work the experiments are performed using RapidMiner 5.3.Two important and well known classification algorithms K-Nearest Neighbor (KNN) and Naive Bayes (NB) are applied to the soil dataset which is taken from the soil testing laboratory Jabalpur, M.P. And classification of soil into low, medium and high categories are done in order to predict the crop yield using available dataset. This study can help the soil analysts and farmers to decide sowing in which land may result in better crop production.

Accuracy: Accuracy is different from both classification methods. By experimenting with KNN algorithm, 30 lands are classified as Low category soil, 45 lands of Medium category soil and 25 lands of High category soil. By experimenting withNaive Bayes algorithm, 15 lands are classified as Low category soil, 40 under Medium category soil and 45 lands of High category soil.

Future Work: In this work, a small dataset is used due to occurrence of some complexities. Thus a larger dataset of 1GB or more may be used in the later work.

2.4 Comparison

	Methods used	Advantages	Disadvantages	Dataset	Result
Approach 1	Random forest,Hadoop framework and MapReduce Programming model.	Large dataset can be easily stored and classified	The performance and accuracy can be increased by reducing work on the classification method.	The dataset is gathered in two ways: Online data and sensor node data.	The proposed work, gives better accuracy and works faster.
Approach 2	CSM method, decision tree, Artificial Neural Networks, Regression.	Sequence of crops that can be grown on the same land is predicted.	If the prediction is not wisely and correctly choosen then there are chances of predicting the wrong sequence of crops.	The crop sowing table, data are gathered from Patna District,Bihar.	The proposed method resolve selection of crops based on yield rate.
Approach 3	Data mining Methods (Naïve Bayes and K-Nearest Neighbors).	Patterns are discovered from the large data set (data regarding the previous years yield, weather etc).	This approach uses small dataset due to some complexities.	The soil dataset is taken from Soil Testing laboratory Jabalpur, MP.	The proposed method helps farmers to decide sowing in which land may result in better crop production

Table 2.1 Comparison of Approaches

Chapter 3. Proposed System

3.1 The Proposal

The proposal is to deploy a system at the entry gate which can analyse soil and predict crop according to soil conditions.

3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system:

- **Economic**: The proposed system is economic as there is no cost involved in it.
- **Man Power**: It does not require any person or their efforts to suggest crops to farmers.
- **24 x 7 Availability**: As it is web application, it is always available provided you have internet connection.
- **Promote Healthy lifestyle of crops**: As right fertilisation suggestion is just at a click.

3.3 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.3.1 Technical

As a conglomeration of math, data science and software engineering, for working and understanding of machine learning algorithms three fields are required to be proficient with. They are: Probability, Statistics and Data Modelling.

HARDWARE REQUIREMENT:

- Intel core i3 or more

- Hard disk: 256GB
- RAM: 4GB

SOFTWARE REQUIREMENT:

- Operating System:WINDOWS 8 and above
- Anaconda Navigator
- Python libraries: sklearn, pandas
- Python framework: Flask

3.3.2 Economical

Since the system is completely automated, there is a need of continuous network supply for it to operate 24X7. The React framework used in the system works great and is in most demand.

3.3.3 Operational

The main motto of our system is to reduce the manual efforts of dietician for diet planning. The system is able to do that accurately and efficiently making the system operationally feasible.

3.4 Design Representation

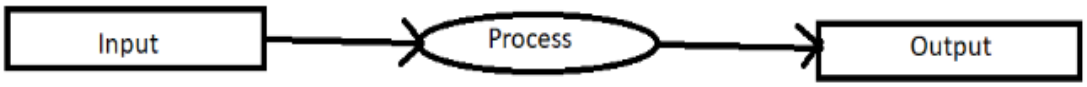


Figure 3-1 Data Flow Diagram Level 0

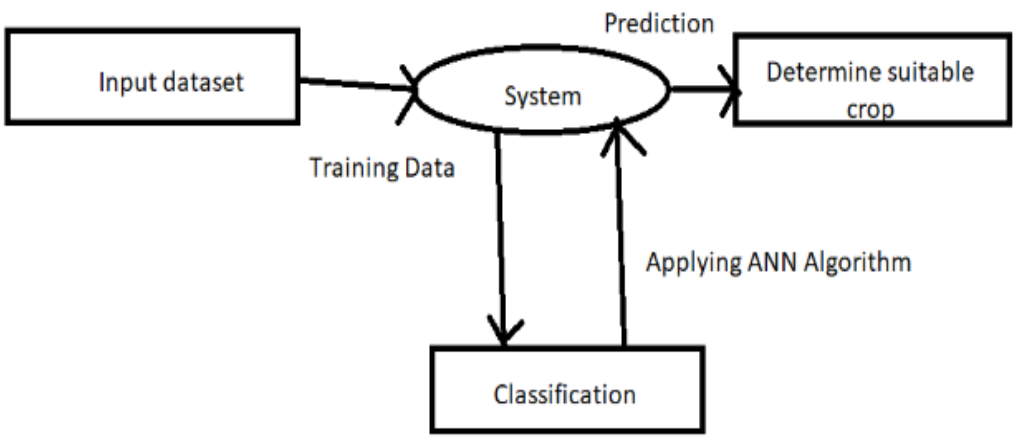


Figure 3-2 Data Flow Diagram Level 1

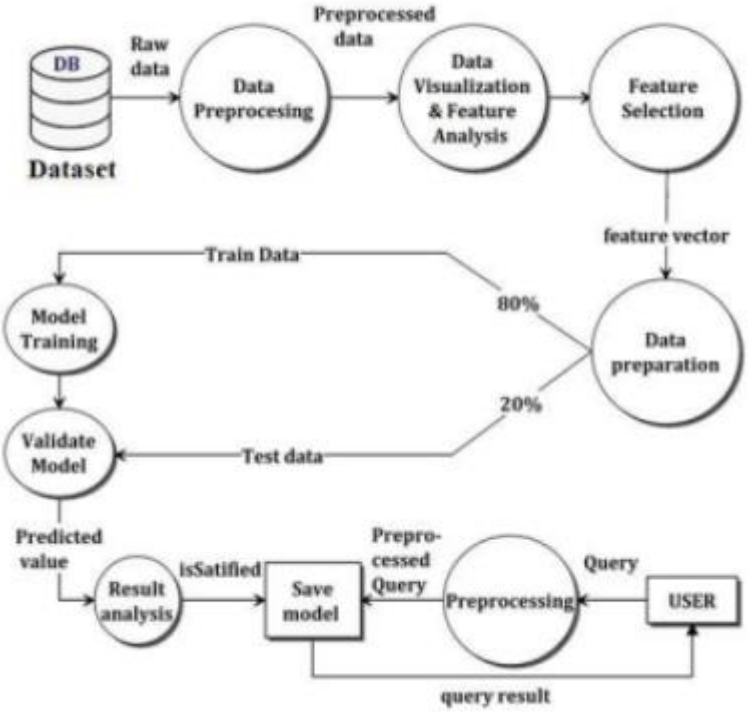


Figure 3-3 Data Flow Diagram Level 2

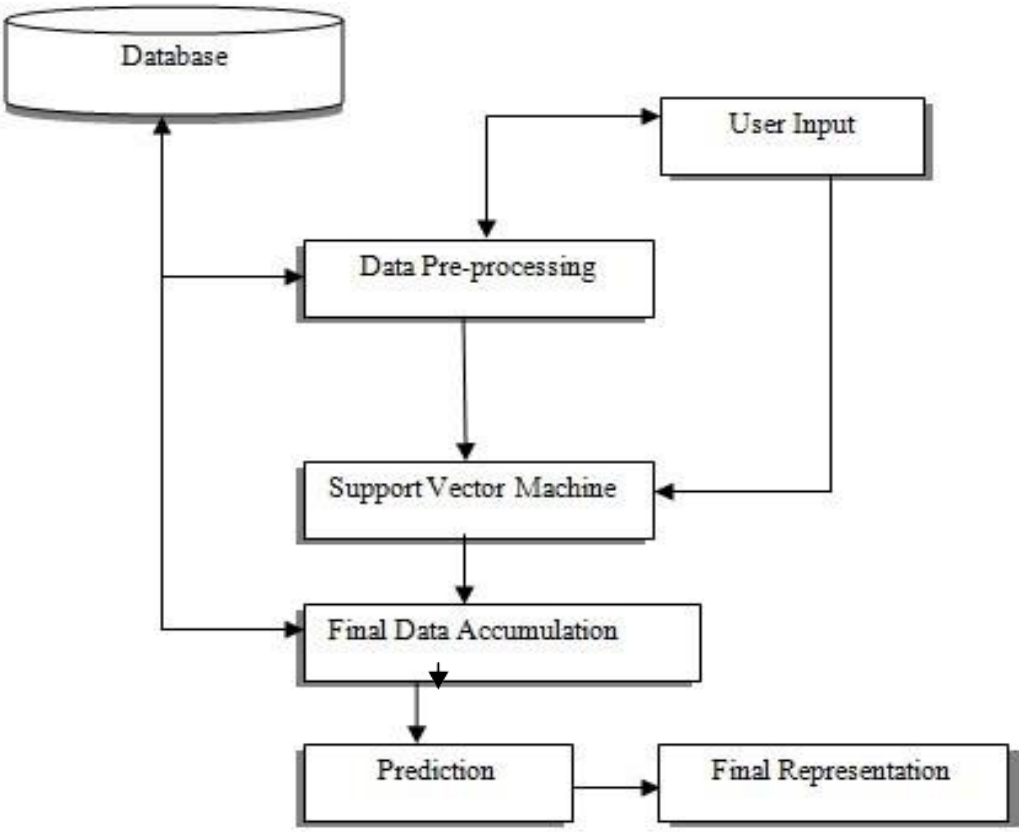


Figure 3-4 Architectural Design

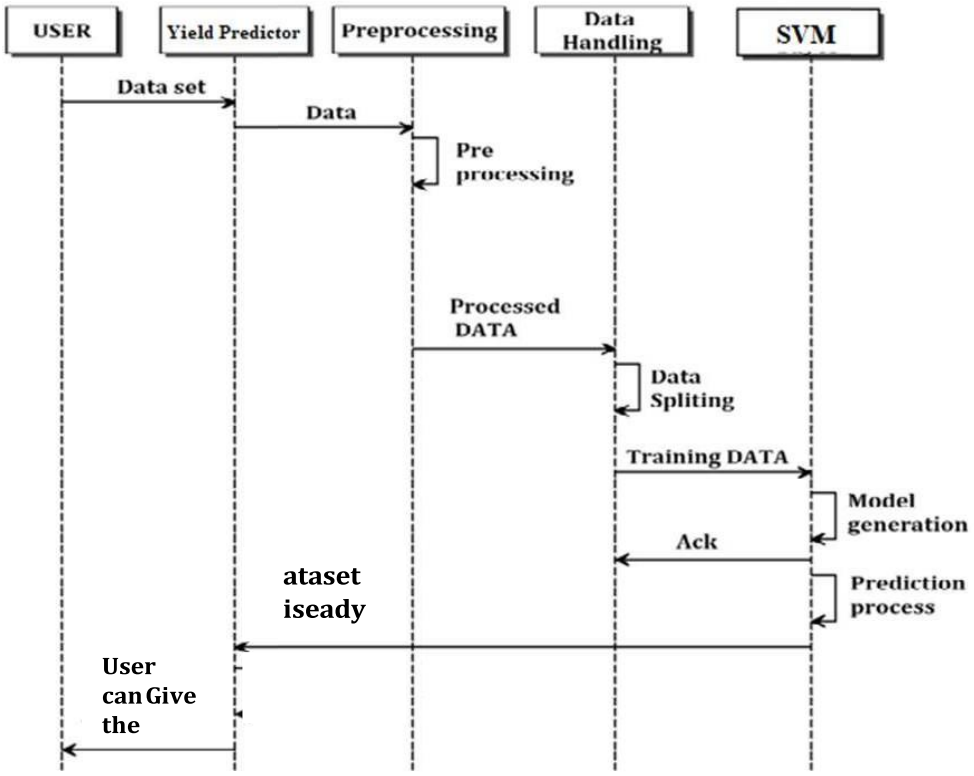


Figure 3-5 Sequence Diagram

3.5 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below:

3.5.1 Hardware

- 32-bit, x86 Processing system
- Windows 7 or later operating system
- High processing computer system

3.5.2 Software

- Python and its supported libraries
- Anaconda

If Installing Flask in VSCODE :

In VS Code, create a new file in your project folder named app.py using either File > New from the menu, pressing Ctrl+N, or using the new file icon in the Explorer View.

In app.py, add code to import Flask and create an instance of the Flask object.

Chapter 4 .Implementation

4.1 Technique and Tools Used

4.1.1 Flask

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Applications that use the Flask framework include Pinterest and LinkedIn

4.1.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

4.2 Language Used

Python language is used in the system due to the following characterstics:

Simple :

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source :

Python is an example of a FLOSS (Free/Libre and Open Source Software).

In simple terms, you can freely distribute copies of this software, read the

software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

Object Oriented:

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

Extensive Libraries:

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML

4.3 Screenshots

The Following are the screenshots of the result of the project:

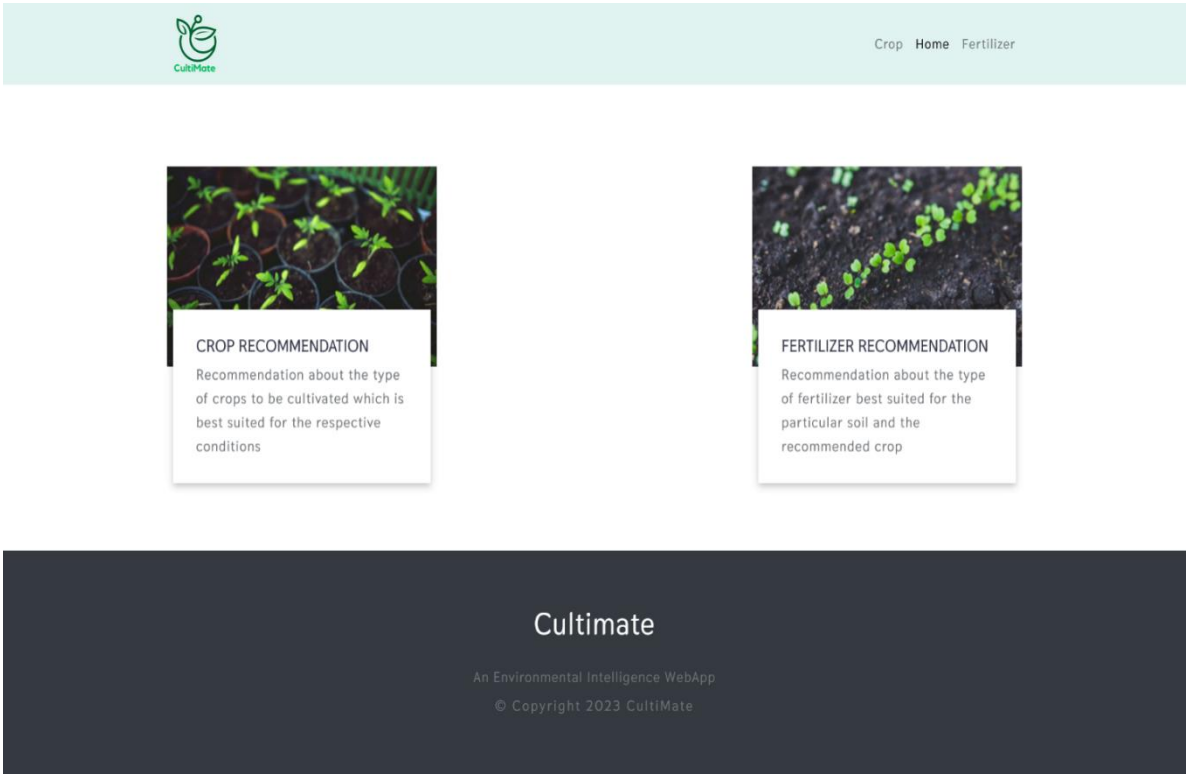


Figure 4-1: Screenshot 1

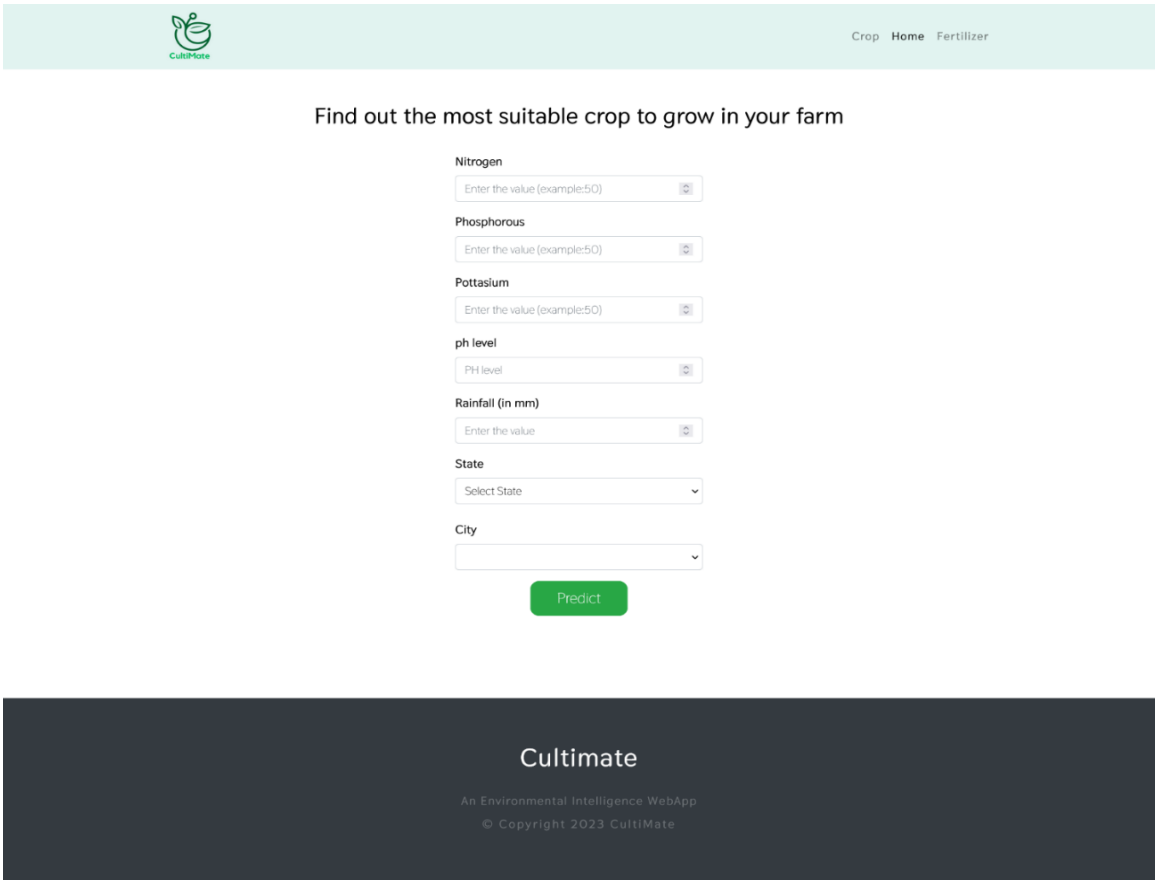


Figure 4-2: Screenshot 2

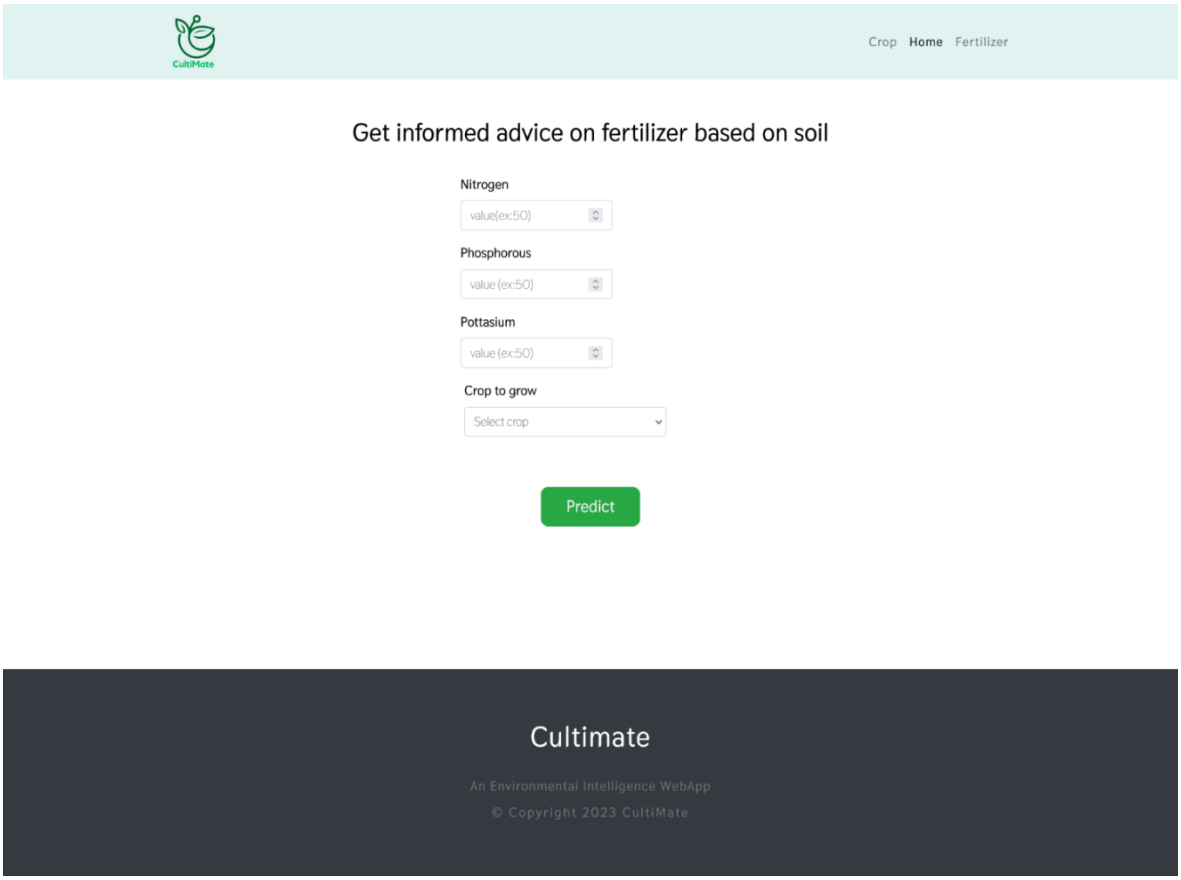


Figure 4-3 Screenshot 3

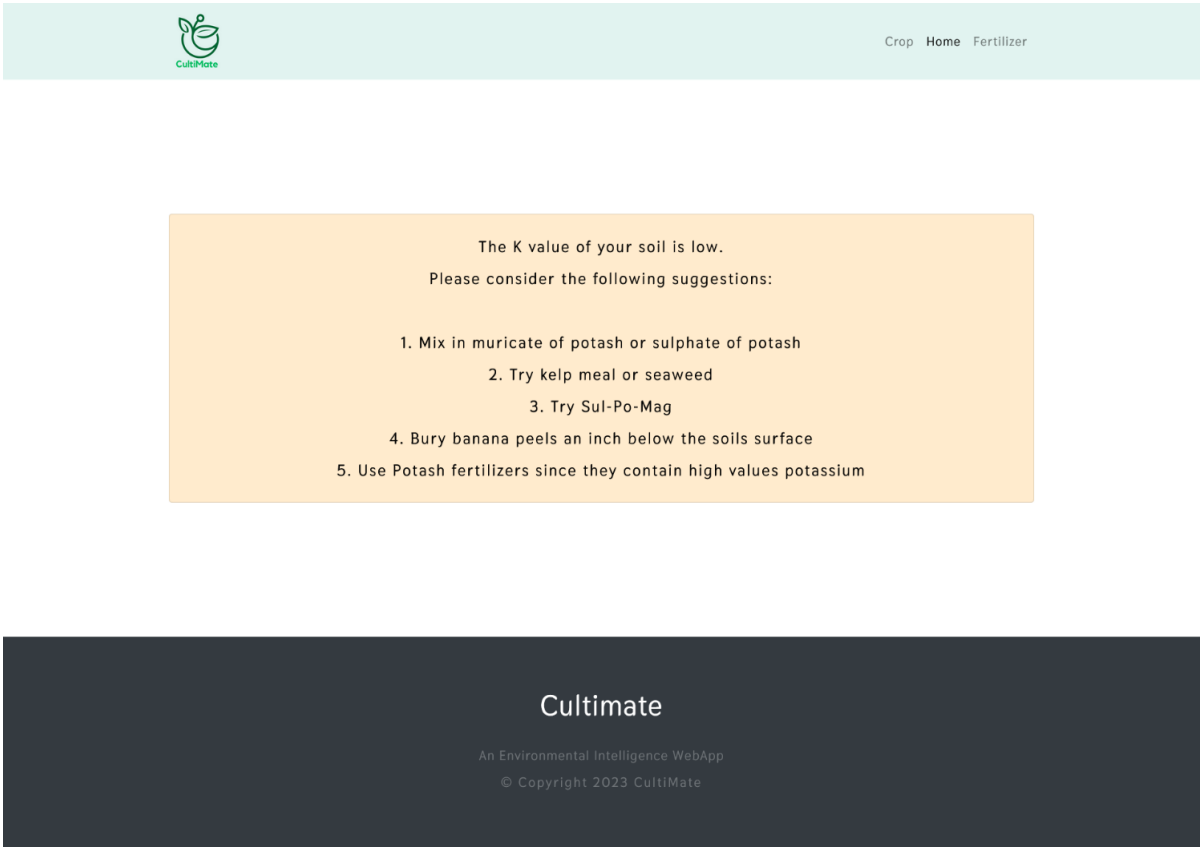


Figure 4-4 Screenshot 4

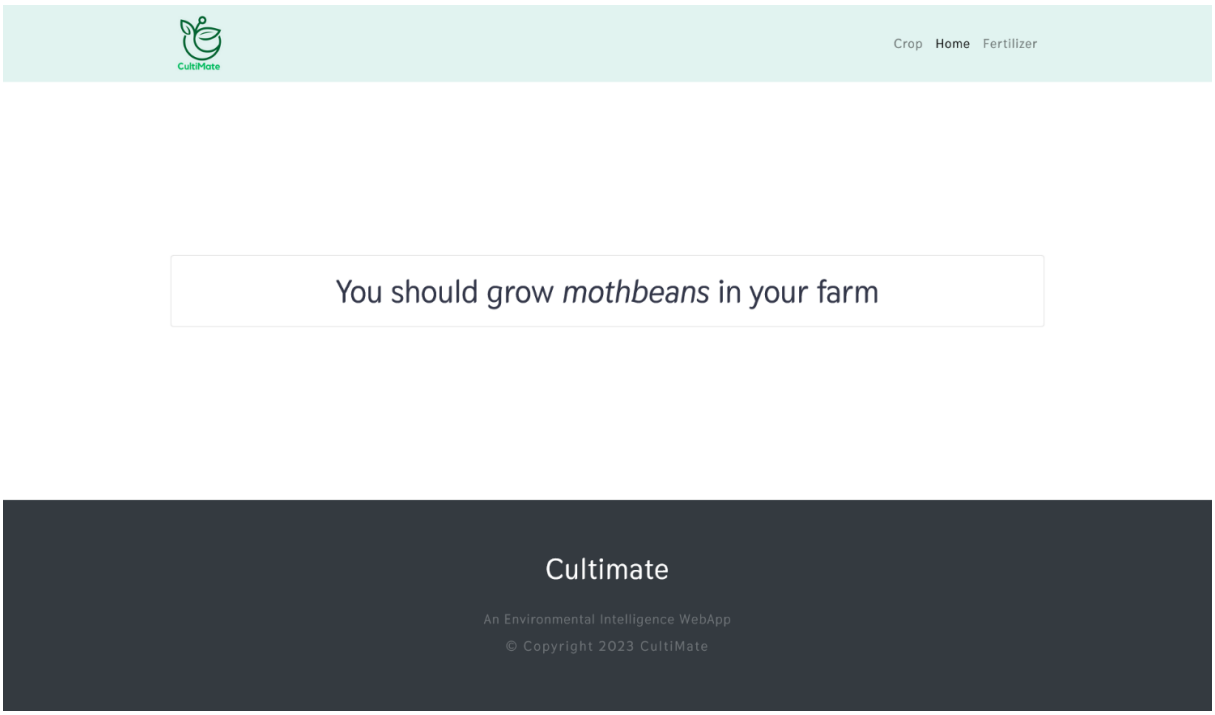


Figure 4-5 Screenshot 5

Chapter 5 . Conclusion

5.1 Conclusion

Agriculture is the backbone of many countries including India. Since integrating the information technology with the agriculture will guide the farmer to improve the productivity. In this proposed work the system described works faster and gives better accuracy in prediction to predict the suitable crops and fertilizers for the field. It includes various parameters of soil to analyse the crop. This prediction makes the farmers to improve the productivity, growth, and quality of the plants.

5.2 Future Scope

In future this project can be modified by adding few more things like:

- Predicting the pesticide ratio based on the atmospheric and soil parameters forthe farm land.
- Database can be created to store the soil details which given by the farmer andalso the predicted crop name.
- New CSV file can be created where the new datasets can be added and thiscan be used for training the model in the future.

Bibliography

1. B Sankara Babu, A Suneetha, G Charles Babu,Y.Jeevan Nagendra Kumar, G Karuna, " Medical Disease Prediction using Grey Wolf optimization and Auto Encoder based Recurrent Neural Network", Periodicals of Engineering and Natural Sciences, June 2018 ISSN 2303-4521 Vol.6, No.1, pp. 229 240
2. Y. Jeevan Nagendra Kumar, Dr. T. V. Rajini Kanth, "GIS MAP Based Spatial Analysis of Rainfall Data of Andhra Pradesh and Telangana States Using R", International Journal of Electrical and Computer Engineering (IJECE), Vol 7, No 1, February 2017, Scopus Indexed Journal, ISSN: 2088-8708
3. International Conference on Computational Intelligence and Communication Networks.
4. N. Heemageetha, "A survey on Application of Data Mining Techniques to Analyze the soil for agricultural purpose", 2016IEEE.
5. M.R. Bendre, R.C. Thool, V.R. Thool, "Big Data in Precision agriculture", Sept, 2015 NGCT.
6. Monali Paul, Santosh K. Vishwakarma, Ashok Verma, "Analysis of Soil Behavior and Prediction of Crop Yield using Data Mining approach", 2015.

Guide Interaction Sheet

Date	Discussion	Action Plan
11/01/2023	Discussed about the title of the project.	Crop Recommendation System was decided as the title.
18/01/2023	Discussion on the technology to be used.	Flask(Python was finalized)
25/01/2023	Discussion of the creation of synopsis of the project.	Gathering of information for synopsis creation
01/02/2023	Suggestions on how to do a literature survey and preliminary investigation on the topic.	Many research papers were read , understood and their abstract were to be written.
08/02/2023	Discussion on the implementation of the project.	Started using Flask framework

15/02/2023	Discussion on the objective of the project.	Decided to Include the logic of crop prediction
22/02/2023	Discussion on project documentation	Decided to write the content and integrate it in the proper format of the report.

Source Code

1. app.py

```
[04:28, 15/04/2023] Ansh Joshi: from flask import Flask, render_template, request, Markup
import numpy as np
import pandas as pd
from utils.disease import disease_dic
from utils.fertilizer import fertilizer_dic
import requests
import config
import pickle
import io
import torch
from torchvision import transforms
from PIL import Image
from utils.model import ResNet9
#
==========================================================================
============================

# ----------------------------LOADING THE TRAINED MODELS -----------------------------------------
-------

# Loading plant disease classification model

disease_classes = ['Apple___Apple_scab',
                   'Apple___Black_rot',
                   'Apple___Cedar_apple_rust',
                   'Apple___healthy',
                   'Blueberry___healthy',
                   'Cherry_(including_sour)___Powdery_mildew',
                   'Cherry_(including_sour)___healthy',
                   'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
                   'Corn_(maize)___Common_rust',
                   'Corn_(maize)___Northern_Leaf_Blight',
                   'Corn_(maize)___healthy',
                   'Grape___Black_rot',
                   'Grape___Esca(Black_Measles)',
                   'Grape___Leaf_blight(Isariopsis_Leaf_Spot)',
                   'Grape___healthy',
                   'Orange___Haunglongbing(Citrus_greening)',
                   'Peach___Bacterial_spot',
                   'Peach___healthy',
                   'Pepper,bell___Bacterial_spot',
                   'Pepper,bell___healthy',
                   'Potato___Early_blight',
                   'Potato___Late_blight',
                   'Potato___healthy',
                   'Raspberry___healthy',
                   'Soybean___healthy',
                   'Squash___Powdery_mildew',
                   'Strawberry___Leaf_scorch',
                   'Strawberry___healthy',
                   'Tomato___Bacterial_spot',
```

```
            'Tomato___Early_blight',
            'Tomato___Late_blight',
            'Tomato___Leaf_Mold',
            'Tomato___Septoria_leaf_spot',
            'Tomato___Spider_mites Two-spotted_spider_mite',
            'Tomato___Target_Spot',
            'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
            'Tomato___Tomato_mosaic_virus',
            'Tomato___healthy']

disease_model_path = 'models/plant_disease_model.pth'
disease_model = ResNet9(3, len(disease_classes))
disease_model.load_state_dict(torch.load(
  disease_model_path, map_location=torch.device('cpu')))
disease_model.eval()


# Loading crop recommendation model

crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
  open(crop_recommendation_model_path, 'rb'))


#
==========================================================================
======================

# Custom functions for calculations


def weather_fetch(city_name):
  """
  Fetch and returns the temperature and humidity of a city
  :params: city_name
  :return: temperature, humidity
  """
  api_key = config.weather_api_key
  base_url = "http://api.openweathermap.org/data/2.5/weather?"

  complete_url = base_url + "appid=" + api_key + "&q=" + city_name
  response = requests.get(complete_url)
  x = response.json()

  if x["cod"] != "404":
    y = x["main"]

    temperature = round((y["temp"] - 273.15), 2)
    humidity = y["humidity"]
    return temperature, humidity
  else:
    return None


def predict_image(img, model=disease_model):
  """
  Transforms image to tensor and predicts disease label
  :params: image
  :return: prediction (string)
  """
```

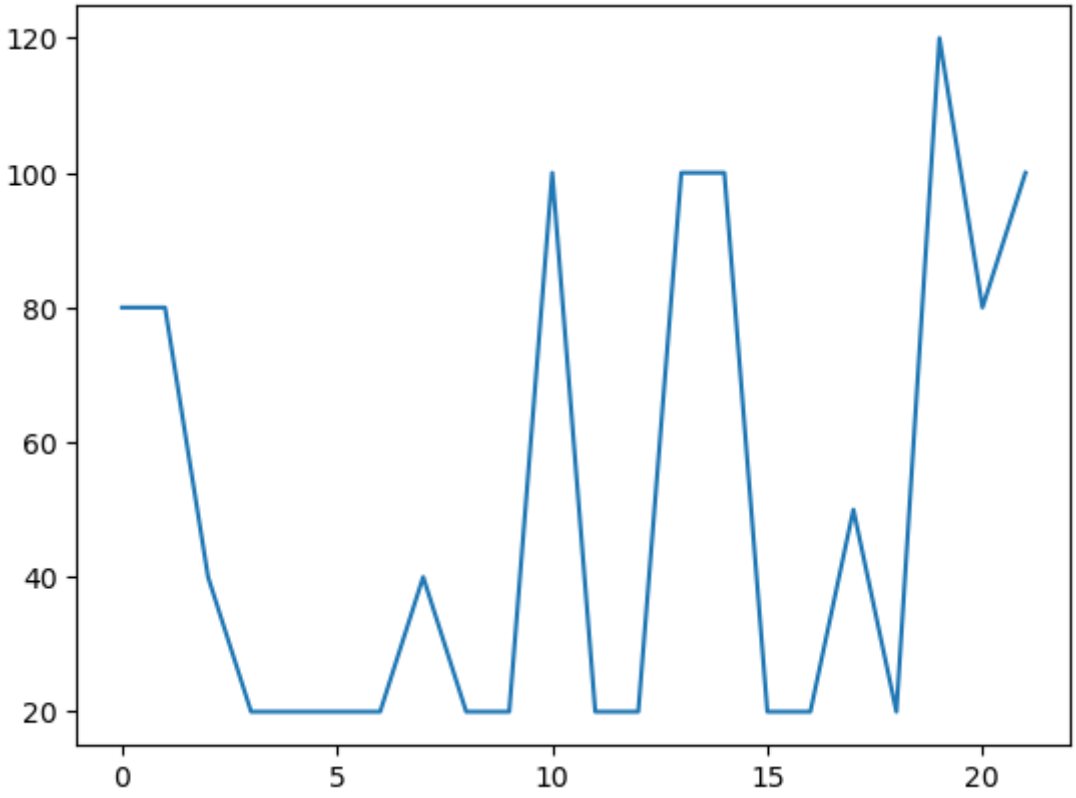
```
    transform = transforms.Compose([
        transforms.Resize(256),
        transforms.ToTensor(),
    ])
    image = Image.open(io.BytesIO(img))
    img_t = transform(image)
    img_u = torch.unsqueeze(img_t, 0)

    # Get predictions from model
    yb = model(img_u)
    # Pick index with highest probability
    _, preds = torch.max(yb, dim=1)
    prediction = disease_classes[preds[0].item()]
    # Retrieve the class label
    return prediction

#
================================================================================
============================
# ----------------------------------- FLASK APP -------------------------------------------------


app = Flask(_name_)

# render home page


@ app.route('/')
def home():
    title = 'Cultimate - Home'
    return render_template('index.html', title=title)

# render crop recommendation form page


@ app.route('/crop-recommend')
def crop_recommend():
    title = 'Cultimate - Crop Recommendation'
    return render_template('crop.html', title=title)

# render fertilizer recommendation form page


@ app.route('/fertilizer')
def fertilizer_recommendation():
    title = 'Cultimate - Fertilizer Suggestion'

    return render_template('fertilizer.html', title=title)

# render disease prediction input page




#
================================================================================
============================

# RENDER PREDICTION PAGES
```

```
# render crop recommendation result page


@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'Cultimate - Crop Recommendation'

    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        if weather_fetch(city) != None:
            temperature, humidity = weather_fetch(city)
            data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
            my_prediction = crop_recommendation_model.predict(data)
            final_prediction = my_prediction[0]

            return render_template('crop-result.html', prediction=final_prediction,
title=title)

        else:

            return render_template('try_again.html', title=title)

# render fertilizer recommendation result page


@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'Cultimate - Fertilizer Suggestion'

    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['pottasium'])
    # ph = float(request.form['ph'])

    df = pd.read_csv('Data/fertilizer.csv')

    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
    elif max_value == "P":
```

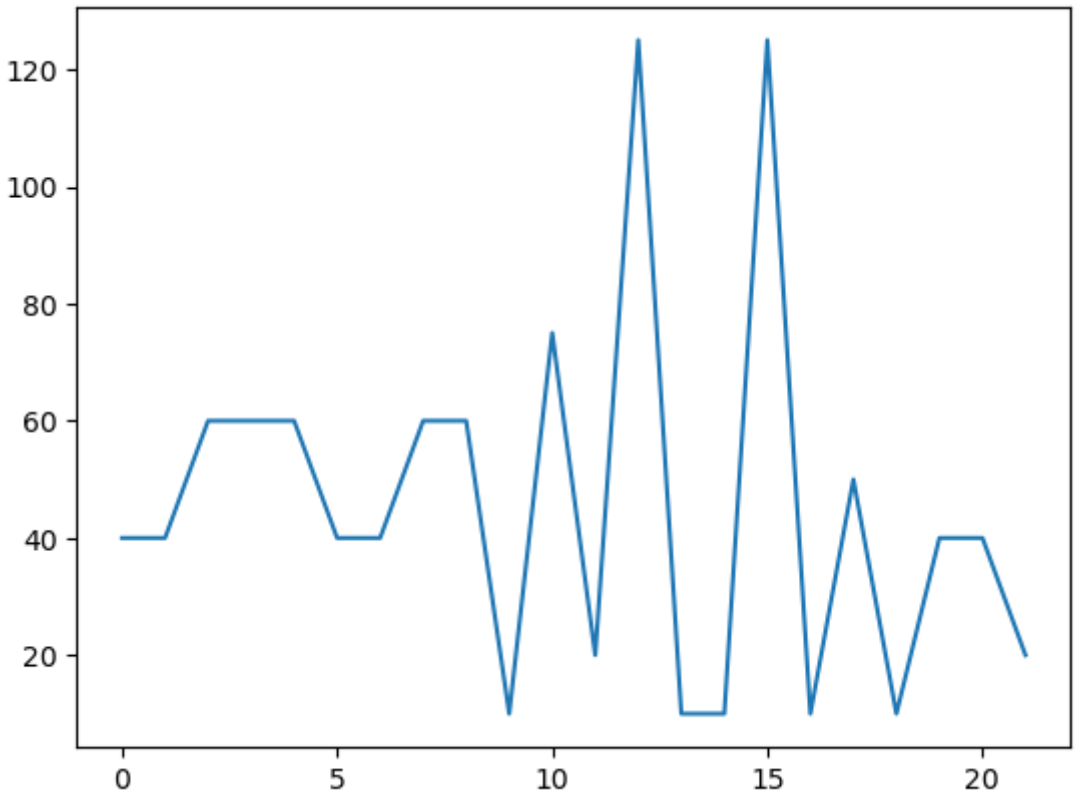
```
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"

    response = Markup(str(fertilizer_dic[key]))

    return render_template('fertilizer-result.html', recommendation=response,
title=title)

# render disease prediction result page


@app.route('/disease-predict', methods=['GET', 'POST'])
def disease_prediction():
    title = 'Cultimate - Disease Detection'

    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files.get('file')
        if not file:
            return render_template('disease.html', title=title)
        try:
            img = file.read()

            prediction = predict_image(img)

            prediction = Markup(str(disease_dic[prediction]))
            return render_template('disease-result.html', prediction=prediction, title=title)
        except:
            pass
    return render_template('disease.html', title=title)


#
==========================================================================
============================
if _name_ == '_main_':
    app.run(debug=False)
 weather_api_key = "9d7cde1f6d07ec55650544be1631307e"
//Config.py
```

1. Models

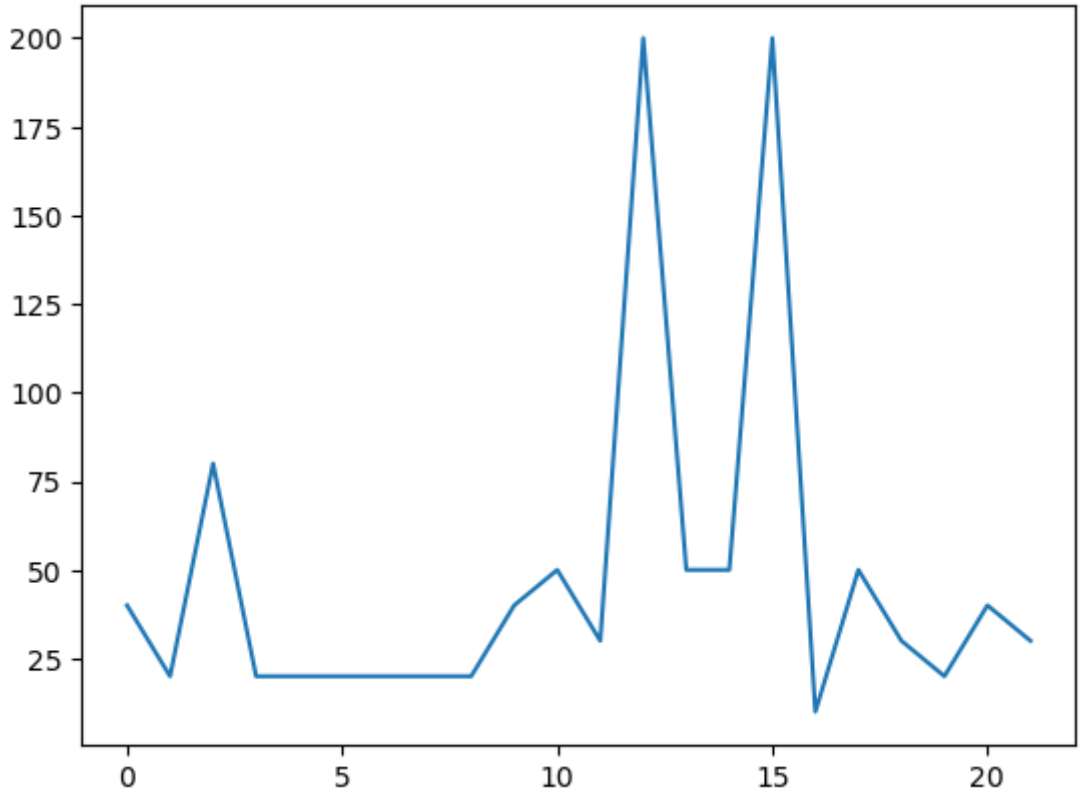
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
fertilizer_data_path = '../Data-raw/FertilizerData.csv'
merge_fert = pd.read_csv(fertilizer_data_path)
merge_fert.head()
del merge_fert['Unnamed: 0']
merge_fert.describe()
merge_fert['Crop'].unique()
plt.plot(merge_fert["N"])
```



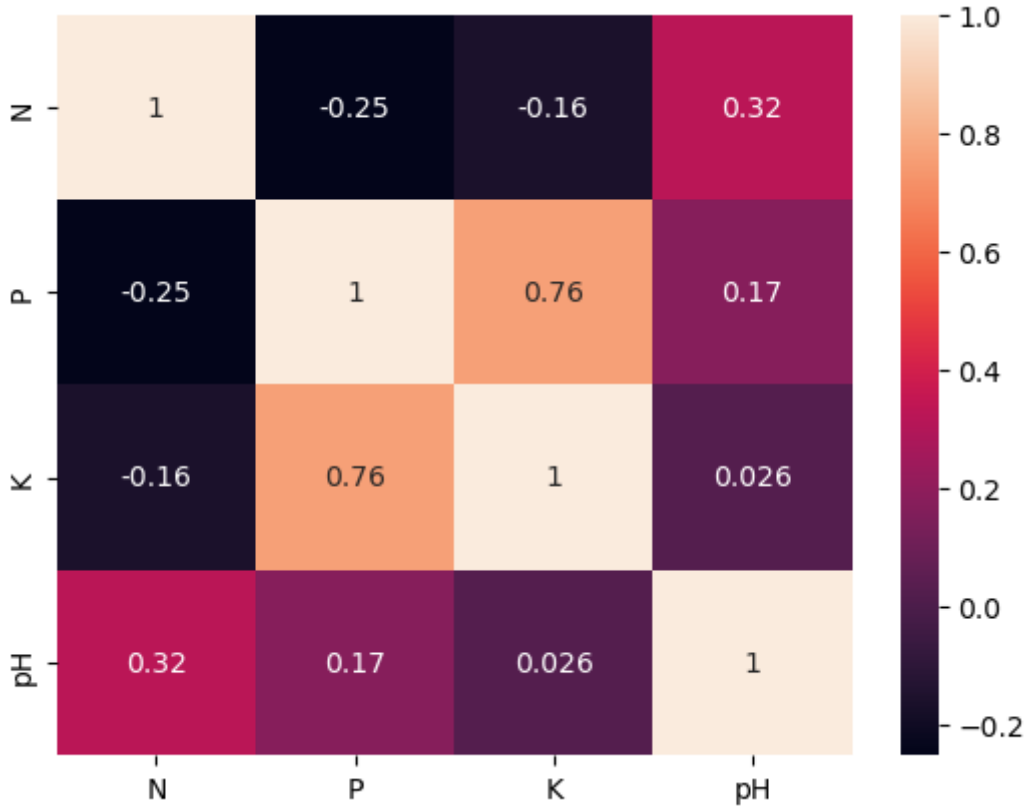

```
plt.plot(merge_fert["P"])
```




```
plt.plot(merge_fert["K"])
```




```
sns.heatmap(merge_fert.corr(),annot=True)
```




```
merge_crop = pd.read_csv('../Data-raw/MergeFileCrop.csv')
reco_fert = merge_fert
#Add +/-3 for every NPK value
import random
temp = pd.DataFrame(columns = ['N','P','K'])
for i in range(0,merge_crop.shape[0]):
    crop = merge_crop.label.iloc[i]
    #print(crop)
    N = reco_fert[reco_fert['Crop'] == crop]["N"].iloc[0] +
random.randint(-20,20)
    P = reco_fert[reco_fert['Crop'] == crop]["P"].iloc[0] +
random.randint(-5,20)
```

```
    K = reco_fert[reco_fert['Crop'] == crop]["K"].iloc[0] +
random.randint(-5,5)
    d = {"N":N,"P":P,"K":K}
    #print(d)
    temp = temp.append(d,ignore_index = True)
temp
```

	N	P	K
0	65	56	35
1	100	59	45
2	68	39	38
3	78	54	41
4	92	57	43
...	...	...	...
2195	94	24	27
2196	118	24	30
2197	94	36	28
2198	104	24	27
2199	80	24	25

```
merge_crop['N'] = temp['N']
merge_crop['P'] = temp['P']
merge_crop['K'] = temp['K']
```

	Unnamed: 0	temperature	humidity	ph	rainfall	label	N	P	K
0	0	20.879744	82.002744	6.502985	202.935536	rice	65	56	35
1	1	21.770462	80.319644	7.038096	226.655537	rice	100	59	45
2	2	23.004459	82.320763	7.840207	263.964248	rice	68	39	38
3	3	26.491096	80.158363	6.980401	242.864034	rice	78	54	41
4	4	20.130175	81.604873	7.628473	262.717340	rice	92	57	43
...	...	...	...	...	...	...	...	...	...
2195	895	26.774637	66.413269	6.780064	177.774507	coffee	94	24	27
2196	896	27.417112	56.636362	6.086922	127.924610	coffee	118	24	30
2197	897	24.131797	67.225123	6.362608	173.322839	coffee	94	36	28
2198	898	26.272418	52.127394	6.758793	127.175293	coffee	104	24	27
2199	899	23.603016	60.396475	6.779833	140.937041	coffee	80	24	25

```
del merge_crop['Unnamed: 0']
merge_crop
```

	temperature	humidity	ph	rainfall	label	N	P	K
0	20.879744	82.002744	6.502985	202.935536	rice	65	56	35
1	21.770462	80.319644	7.038096	226.655537	rice	100	59	45
2	23.004459	82.320763	7.840207	263.964248	rice	68	39	38
3	26.491096	80.158363	6.980401	242.864034	rice	78	54	41
4	20.130175	81.604873	7.628473	262.717340	rice	92	57	43
...	...	...	...	...	...	...	...	...
2195	26.774637	66.413269	6.780064	177.774507	coffee	94	24	27
2196	27.417112	56.636362	6.086922	127.924610	coffee	118	24	30
2197	24.131797	67.225123	6.362608	173.322839	coffee	94	36	28
2198	26.272418	52.127394	6.758793	127.175293	coffee	104	24	27
2199	23.603016	60.396475	6.779833	140.937041	coffee	80	24	25

```
merge_crop = merge_crop[[ 'N', 'P', 'K','temperature',
'humidity', 'ph', 'rainfall', 'label']]
merge_crop.to_csv("../Data-processed/crop_recommendation.csv",index=False)
# Checking if everything went fine
df = pd.read_csv('../Data-processed/crop_recommendation.csv')
```