A Project Report

On

GROWSMART- AN ML BASED INTELLIGENT RECOMMENDATION SYSTEM FOR CROPS AND FERTILIZERS

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In Partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted By

R.M SAI MOHIT	**- (19699A0544)**
M. MADHUSUDHAN NAIDU	**- (19699A0524)**
B. JAGADEESH NAIK	**- (19699A0517)**
K.V HARIPRASAD REDDY	**- (19699A0513)**

Under the Guidance of

DR. S. KUSUMA, M.TECH, PH.D

Assistant Professor

Department of Computer Science & Engineering

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

Accredited by NBA, Approved by AICTE, New Delhi)

AN ISO 9001:2015 Certified Institution

P. B. No: 14, Angallu, Madanapalle – 517325

2019-2023

✆ : 08571 - 280255, 280590

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

AN ENGINEERING COLLEGE SPONSORED BY :
RATAKONDA RANGA REDDY EDUCATIONAL ACADEMY
Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapur
P.B. No. 14, Angallu, MADANAPALLE - 517 325. Chittoor Dist. A.P.
www.mits.edu

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"GROWSMART- AN ML BASED INTELLIGENT RECOMMENDATION SYSTEM FOR CROPS AND FERTILIZERS"** is a bonafide work carried out by

R.M. SAI MOHIT	**- (19699A0544)**
M. MADHUSUDHAN NAIDU	**- (19699A0524)**
B. JAGADEESH NAIK	**- (19699A0517)**
K.V HARIPRASAD REDDY	**- (19699A0513)**

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science & Engineering** in **Madanapalle Institute of Technology and Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2022-2023

Guide
Dr. S. Kusuma, M.Tech, Ph.D
Assistant Professor,
Department of CSE

Head of the Department
Dr. R. Kalpana, Ph.D
Professor and Head,
Department of CSE

Submitted for the University examination held on:

Internal Examiner
Date:

External Examiner
Date:

ACKNOWLEDGEMENT

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D**., **Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology and Science,** Madanapalle.

We express our deep sense of gratitude to **Dr. R. Kalpana, M. Tech., Ph.D., Professor and Head of the Department of CSE** for her continuous support in making necessary arrangements for the successful completion of the Project.

We express my deep sense gratitude to **Dr. K P Manikandan, Ph.D, Project Coordinator and Mrs. M Bommy M.E., Project Co Coordinator** for their valuable guidance and encouragement that helped us to complete this project.

We express our deep gratitude to my guide **Dr. S. Kusuma**, **M.Tech, Ph.D Assistant Professor, Department of CSE** for her guidance and encouragement that helped us to complete this project.

We also wish to place on record my gratefulness to other **Faculty of CSE Department** and to our friends and our parents for their help and cooperation during our project work.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS INSTITUTION)

Affiliated to JNTUA, Ananthapuramu & Approved by AICTE, New Delhi

NAAC Accredited with A+ Grade, NIRF India Rankings 2021 - Band: 201-250 (Engg.)

NBA Accredited - B.Tech. (CIVIL, CSE, ECE, EEE, MECH), MBA & MCA

RECOGNISED RESEARCH CENTER

Plagiarism Verification Certificate

This is to certify that the B. Tech Project report titled, "**GROWSMART- AN ML BASED INTELLIGENT RECOMMENDATION SYSTEM FOR CROPS AND FERTILIZERS**" submitted by **R. M. SAI MOHIT (REGD. NO: 19699A0544), M. MADHUSUDHAN NAIDU (REGD. NO: 19699A0524), B. JAGADEESH NAIK (REGD. NO: 19699A0517) and K. V. HARIPRASAD REDDY (REGD. NO: 19699A0513)** has been evaluated using **Anti-Plagiarism Software, URKUND** and based on the analysis report generated by the software, the report's similarity index is found to be 8**%**.

The following is the URKUND report for the project report consisting of 44 pages.

Document Information

Analyzed document	19699A0544_SAI MOHIT. R M_Ver1.docx (D165956960)
Submitted	2023-05-05 07:17:00
Submitted by	kusuma
Submitter email	kusumas@mits.ac.in
Similarity	8%
Analysis address	kusumas.mits@analysis.urkund.com

GUIDE

DECLARATION

We hereby declare that the results embodied in this project **"GROWSMART- AN ML BASED INTELLIGENT RECOMMENDATION SYSTEM FOR CROPS AND FERTILIZERS"** by us under the guidance of **Mrs. S. Kusuma**, **M.Tech, Ph.D, Assistant Professor, Dept. of CSE** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science & Engineering** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and we have not submitted the same to any other University/institute for award of any other degree.

Date :

Place :

PROJECTASSOCIATES

R. M. SAI MOHIT

M. MADHUSUDHAN NAIDU

B. JAGADEESH NAIK

K.V HARIPRASAD REDDY

I certify that above statement made by the students is correct to the best of my knowledge.

Date : **Guide**

TABLE OF CONTENTS

CHAPTER 5 IMPLEMENTATION AND RESULTS

CHAPTER 6 TESTING

CHAPTER 7 CONCLUSION

ABSTRACT

Agriculture is a major source of income and employment in India. The most prevalent problem faced by Indian farmers is that they do not select the appropriate crop for their land and do not use the appropriate fertilizer. They will experience a significant drop in production as a result of this. GrowSmart has been used to solve the farmers difficulties. GrowSmart is a modern farming strategy that employs research data on soil properties, soil types, and crop yield statistics to recommend the best crop to farmers as well as fertilizer recommendations. This decreases the number of times a crop is chosen incorrectly and increases productivity. The aim of this analysis is to develop a system which helps the farmers to recommend a crop for the site-specific parameters with high accuracy and efficiency with the help Machine Learning (ML) approaches.

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 Motivation

Agriculture is one of the biggest sources of earning for Indians. Though the Indian farmers really work hard in their fields, their productivity is threatened by natural factors. The fact cannot be changed that natural factors are uncontrollable so the best way is to take most out of the field despite the natural factors. One of the major problems is soil degradation which can be prevented by growing the crop that is the most suitable as per the land. But even if the farmer chooses to grow a particular type of crop, then the appropriate dosage of fertilizers would help. Another major problem is the pest which can be only treated through suitable pesticides. This will help farmers. Various tests are conducted by the government in India which check the contents of soil but farmers are unaware of what to do with the results of the soil test. Hence **GROWSMART** makes use of all the values of the tests and helps the farmers with crop, fertilizer, and pesticide recommendations respectively.

1.2 Problem Definition

GrowSmart wants to assist Indian farmers and lessen their suffering. The issues Indian farmers confront can be summed up as follows:

1. Productivity needs to be raised so farmers can earn more money from the same plot of land while preserving the soil.
2. Indian farmers cannot choose the right crop for their soil's requirements, which depend on parameters like pH, N, P, and K, as well as temperature, humidity, and rainfall.
3. Farmers typically do not know whether organic or conventional fertilizers to use based on the needs of the soil.
4. Due to inadequate and unequal fertilization, soil deterioration is occurring, which promotes nutrient mining and the establishment of second-generation nutrient management problems.

1.3 Objective Of The Project

- To build a robust model to provide correct and accurate prediction of crop sustainability for the soil type and climatic conditions.
- To build user friendly and interactive application.
- Provide recommendation of the most effective suitable crops within the area in order that the farmer does not incur any losses.
- Provide fertilizer suggestion and recommendations based on the crop and N, P, and K values.
- Provide disease prediction and Pesticide recommendation System for crops.
- To identify the pest and suggest a specific pesticide that is offered in India in accordance with ISO criteria.
- To build a best fertilizer and pesticide store.

1.4 Organization Of Chapters

1.4.1 Feasibility Study

Fundamental examination looks at project practicality; the probability the framework will be valuable to the association. The primary goal of the practicality review is to test the Technical, Operational and Economical achievability for adding new modules and investigating old running framework. All frameworks are possible in the event that they are given limitless assets and endless time. There are angles in the practicality concentrate on part of the starter examination:

- Technical Feasibility
- Operation Feasibility
- Economic Feasibility

1.4.1.1 Technical Feasibility

The specialized issue normally raised during the attainability phase of the examination incorporates the accompanying:

- Does what is presented already have the necessary basic innovation?

- Do the proposed gear 's have the specialized ability to hold the information expected to utilize the new framework.
- Will the proposed framework give sufficient reaction to requests, no matter what the number or area of clients?
- Are there specialized assurances of exactness, dependability, simple entry and information security?

1.4.1.2 Operation Feasibility

The operational feasibility includes User friendly, reliability, security, portability, availability, and maintainability of the software used in the project.

1.4.1.3 Economic Feasibility

Analysis of a project costs and revenue to determine whether it is logical and possible to complete.

CHAPTER-2
LITERATURE SURVEY

2.1 Introduction

India relies heavily on agriculture as a source of income, and Indian farmers work tirelessly to provide for their families. In general, farmers deal with pests, pesticides, fertilizers, and crops. In order to assist Indian farmers, GrowSmart offers the Crop Recommendation, Fertilizer Recommendation, and Pesticide Recommendation modules. The field of crop recommendation has been extensively researched, yet the parameters that are given into the ML model differ across all systems. Most ML models employ Random Forest, while some make use of Decision Tree and Ensemble techniques via Majority Voting Mechanism. In the field of AI, fertilizer recommendation is not very effective. The main cause may be corrupted data, however GrowSmart gathered all the information from many sources and combined it to create a well-formed dataset. GrowSmart uses a dictionary-based approach to solving the problem. Thirdly, researchers have solely focused on pest detection; GrowSmart extends the concept of pest identification and provides a dictionary-based solution for the appropriate insecticide, which is readily available in India. For pesticide recommendations, GrowSmart follows the ISO 9001, ISO 14001, and ISO 17025 standards. The difficulty is that searching there is difficult and the majority of pesticides recommended aren't available in India. The majority of pesticides are taken from the biostadt site, which is a fairly popular site for farmers. The research papers related to the services provided by GrowSmart are detailed below.

The paper " Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers " served as our foundation.

(Rajak et al. 951–952), i.e. paper [1], discusses crop prediction using a variety of learners, including Multilayer Perceptron (ANN), Random Forest, Naive Bayes, and SVM utilized as a classifier. pH, depth, water holding capacity, drainage, and erosion are the factors considered when predicting crops.

(Dighe et al. 476-480), i.e., study [2] examined the CHAID, KNN, K-means, Decision Tree, Neural Network, Naive Bayes, C4.5, LAD, IBK, and SVM algorithms and produced rules for recommendation systems. The most likely crops for planting were chosen by considering different aspects, including the pH level of the soil, the month of culture, the local climate, temperature, and soil type.

Location detection, data analysis and storage, similar location detection, and a module for recommendation creation were all covered in (Mokarrama and Arefin) i.e., article [3]. The final crop was obtained using the physiographic database, thermal zone database, crop growth period database, crop production rate database, and seasonal crop database.

In their study [4] (Gandge and Sandhya), they discuss attribute selection, multiple linear regression, decision trees utilizing ID3, SVM, neural networks, C4.5, K-means, and KNN. The suggested system starts by choosing an agricultural field, then choosing a crop that has already been planted. It receives input from the user, preprocesses it, then chooses an attribute in the backend, then applies a classification algorithm to the data, and finally suggests a crop.

(Mishra et al.) That is, paper [5] employs the J48, LAD Tree, LWL, and IBK algorithms; first, WEKA tool is used. LAD Tree demonstrated the lowest accuracy, but trimming the tree can reduce the mistakes, IBK provided good accuracy.

(Wu et al.) The IP102 dataset, which contains about 75, 000 images of 102 insects, is a large-scale dataset for the identification of insect pests. The IP102 complies with a variety of characteristics of insect pest dispersal in real-world situations when compared to prior datasets. In the meantime, they test various cutting-edge recognition methods using the dataset. The results demonstrate that current deep feature methods and handmade feature methods are insufficient for identifying pests.

(Kasinathan et al.) study, which is [7] Different insect datasets were recognized and detected using a machine learning and insect pest detection method, and the outcomes were associated. The CNN model, ANN, SVM, KNN, Naive Bayes, and SVM models were used to compare classification accuracy of various machine learning methods. The results show that the CNN model, using the Wang and Xie datasets, has the best classification precision, with 91.5 percent and 90 percent for 9 and 24 insect groups, respectively.

Using images from traps, a technique for autonomously tracking pests is described in the publication by Ding and Taylor (i.e., document [8]). A convolutional neural network is applied to image patches at different points in a sliding window-based detection pipeline that is claimed to reduce the likelihood of owning a specific pest species. Image patches are thresholded and non-maximum suppressed depending on their placements and associated confidences to produce the final detections.

Paper (TÜRKOLU and HANBAY) [9] Deep feature extraction and transfer learning's various effects on the detection of plant diseases and pests are contrasted. These deep models' deep features for tuning layers were recovered. SVM, ELM, and KNN classifiers were then used to identify the outcomes of the acquired deep features. Using images of plant diseases and pests, deep models were then adjusted. According to the evaluation's findings, deep learning models produced better results than traditional methods. Deep feature extraction produced more accurate results than transfer learning did.

Paper [10] by Selvaraj et al. massive datasets of expert-prescreened photos of banana pest and disease symptoms and damage were acquired using a transfer learning approach, and they were then used to construct three different convolutional neural network (CNN) designs for identification. A simple and effective method for spotting pests and diseases in digital bananas is the DCNN. ResNet50 and InceptionV2 based models outperformed MobileNetV1 when using deep transfer learning (DTL), which uses a pre-trained disease recognition algorithm to build a network that can generate accurate predictions.

2.2 Existing System

- Existing system are restricted to few algorithms only.
- Highest accuracy of existing system is Random Forest which ranges from 90-95 %
- The User alone can see the crops that are recommended based on the input.
- Existing system uses only few parameters.
- Existing system does not contain Pesticide Recommendation System and Store.

2.3 Disadvantages Of Existing System

- No User Interface.
- Only restricted to few algorithms.
- Trained and tested with only few crops.

2.4 Proposed System

- We are going to use 7 best machine learning algorithms and we will take the best algorithm for recommendations.
- After finding the highest accuracy algorithm that is used for testing new data.
- New data is entered using Web Interface and it will recommend the crops, fertilizers and pesticides based on the input and provides suggestions.
- The user can visit Fertilizer & Pesticide store.

2.5 Advantages Of Proposed System

- Easy to Use and Easy to Understand.
- User Friendly application.
- Works efficiently for huge datasets.
- User can avail multiple services for crops, fertilizers, and pesticides.

CHAPTER 3
ANALYSIS

3.1 Introduction

A Software Requirements Specification (SRS) - a necessities particular for a product framework - is a finished portrayal of the way of behaving of a framework to be created. It incorporates a bunch of purpose cases that depict every one of the communications the clients will have with the product. Notwithstanding the utilization cases, the SRS likewise contains non-utilitarian necessities. Non-useful necessities are prerequisites which force imperatives on the plan or execution, (for example, operational efficiency necessities, quality norms, or plan requirements).

3.2 Requirement Specifications

3.2.1 Hardware Requirements

1	RAM	8GB/16GB for faster output
2	Hard Disk	1TB
3	Micro Processor	2.0 GHz
4	Micro Processor Type	Core i5/core i7
5	Any Desktop or Laptop systems with high level configuration	

Table 3.2.1: Hardware Requirements

3.2.2 Software Requirements

1	OS (Operating Systems)	Windows 10/11
2	Programming Language	Python, HTML, CSS
3	IDE (Integrated Development Environment)	Anaconda Navigator -Jupyter notebook, PyCharm
4	APIs	NumPy, Pandas, Sklearn, Matplotlib, Seaborn, Flask
5	User Interface	Flask
7	Technology Used	Machine Learning

Table 3.2.2: Software Requirements

CHAPTER 4
DESIGN

4.1 UML Diagrams

UML represents Unified Modeling Language. The goal of UML to made into a typical language for developing models of article arranged PC programming. In its present structure, UML is contained two significant segments: A Meta-model and a documentation. Later, some type of technique or interaction may likewise be added to; or related with, UML. The Unified Modeling Language is a standard language for determining, Visualization, Constructing, and archiving the antiques of a product frameworks, just as for business displaying and other non-programming frameworks. The UML addresses an assortment of best designing practices that have demonstrated effective in the displaying of enormous and complex frameworks. The UML plays an important role in product improvement measure and creating object arranged programming

Goals Of UML

- It is a general-purpose modeling language.
- Due to the lack of standardized procedures at the time, UML was created after the establishment of object-oriented concepts to systematize and consolidate object-oriented development.
- The UML diagrams are created for business customers, developers, regular folks, or anybody else wishing to comprehend the system, regardless of whether it is composed of software or not.

Characteristics Of UML

The UML has the following features:

- It is a generalized modeling language.
- It is different from other programming languages like C++, Python, etc.
- It is related to design and object-oriented analysis.

4.1.1 Use Case Diagram

The dynamic behaviour of a system is represented by a use case diagram. For the system's functionality to be fully encapsulated, it includes use cases, actors, and their interactions. A system or application subsystem's needs are simulated in terms of tasks, services, and operations. It depicts the high-level functioning of a system as well as how a user engages with that system.

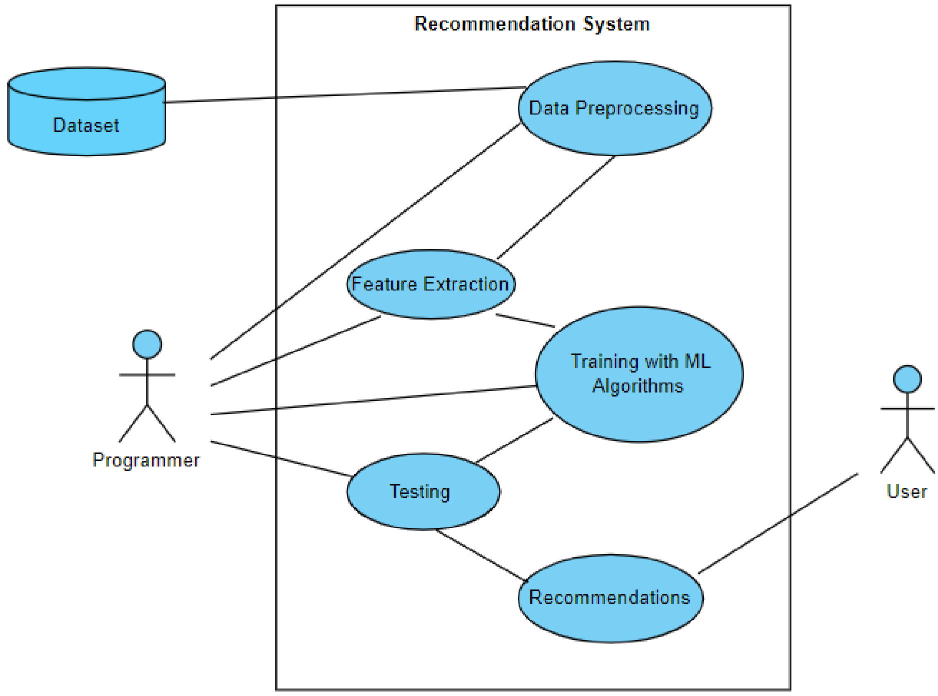


Fig 4.1.1.1: Use Case Diagram from programmer's end

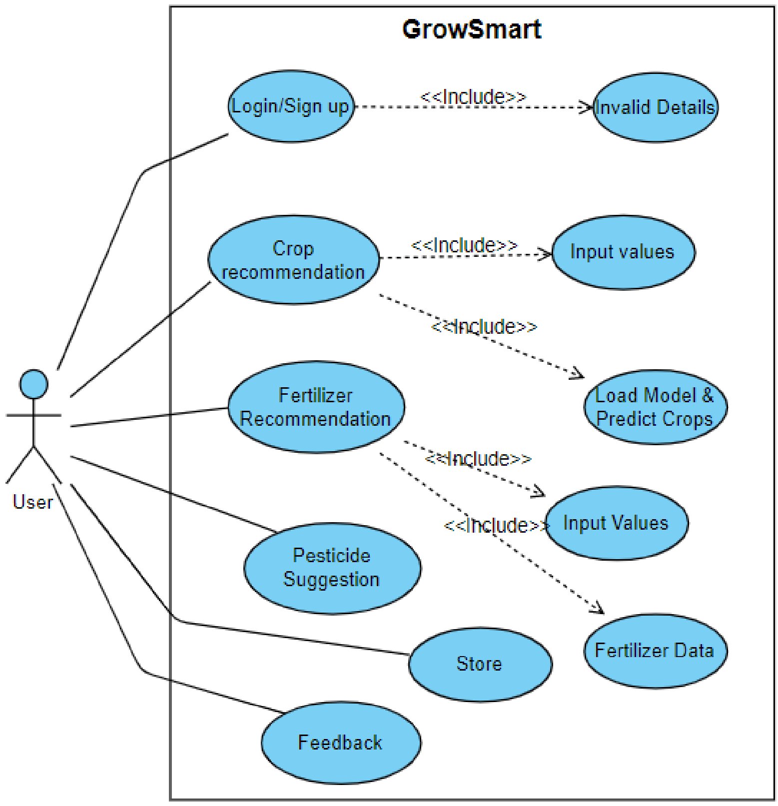


Fig 4.1.1.2: Use Case Diagram from user's end

4.1.2 Class Diagram

A static perspective of an application is depicted in the class diagram. It depicts the different kinds of objects that are present in the system as well as their interactions. In addition to having its own objects, a class can also inherit from other classes. Various distinct aspects of the system are visualised, described, documented, and executable software code is also created using class diagrams.

It displays the classes, relationships, properties, and functions to provide a summary of the software system. In a separate section, it organises class names, characteristics, and functions to aid in software development.

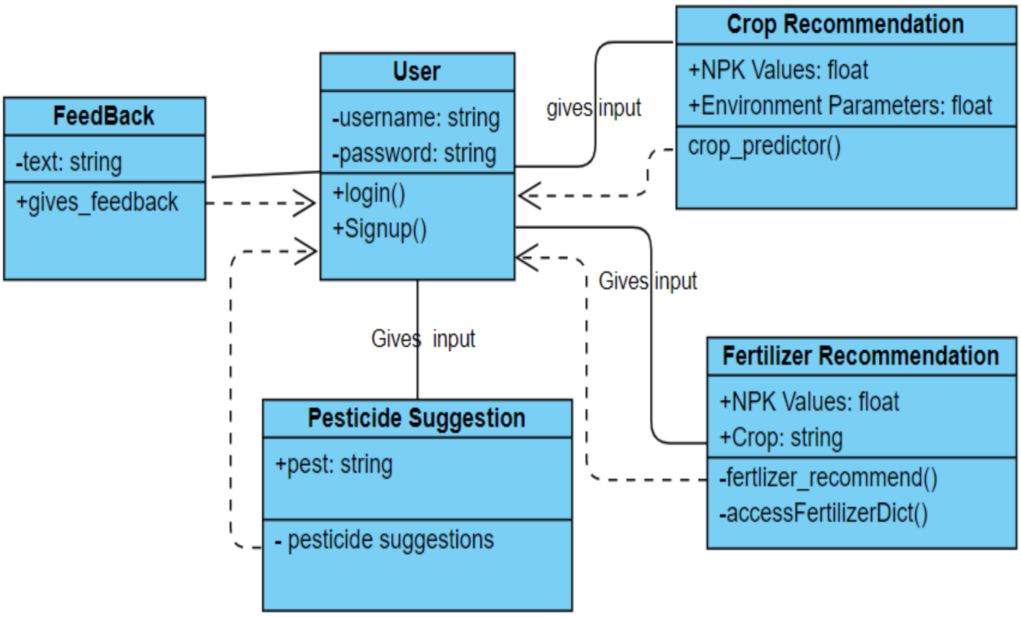


Fig 4.1.2 – Class Diagram

4.1.3 Activity Diagram

Instead of demonstrating implementation, the activity diagram is used to show how control flows within the system. It simulates both sequential and concurrent tasks.

The activity diagram makes it easier to visualise the flow of one action into the next. It highlighted both the state of flow and the sequence in which it takes place. The flow may be concurrent, branching, or sequential, and the activity diagram has been designed with a fork, join, etc. to deal with these many types of flows.

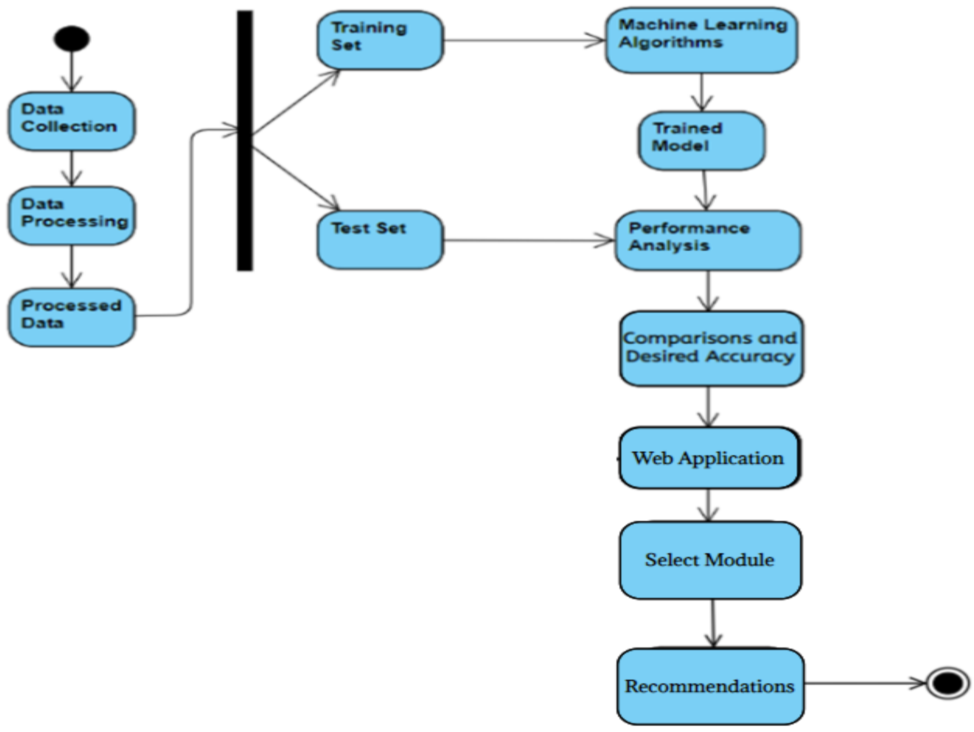


Fig 4.1.3.1: Activity Diagram 1

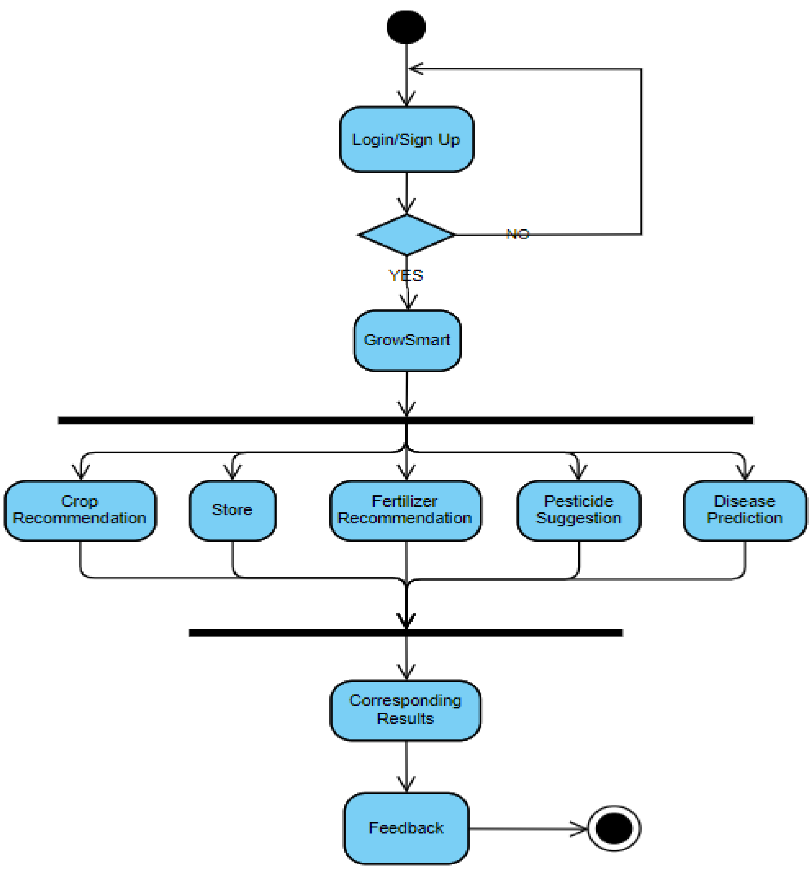


Fig 4.1.3.2: Activity Diagram 2

4.1.4 Sequence Diagram

The sequence diagram, also known as an event diagram, depicts the flow of messages through the system. Several dynamic scenarios can be imagined with its assistance. It depicts the interaction of any two lifelines as a time-ordered series of activities in which they were involved at the time of the communication. The message flow is represented by a vertical dotted line that runs across the bottom of the page in UML, whereas the lifeline is represented by a vertical bar. As well as branching, it incorporates iterations.

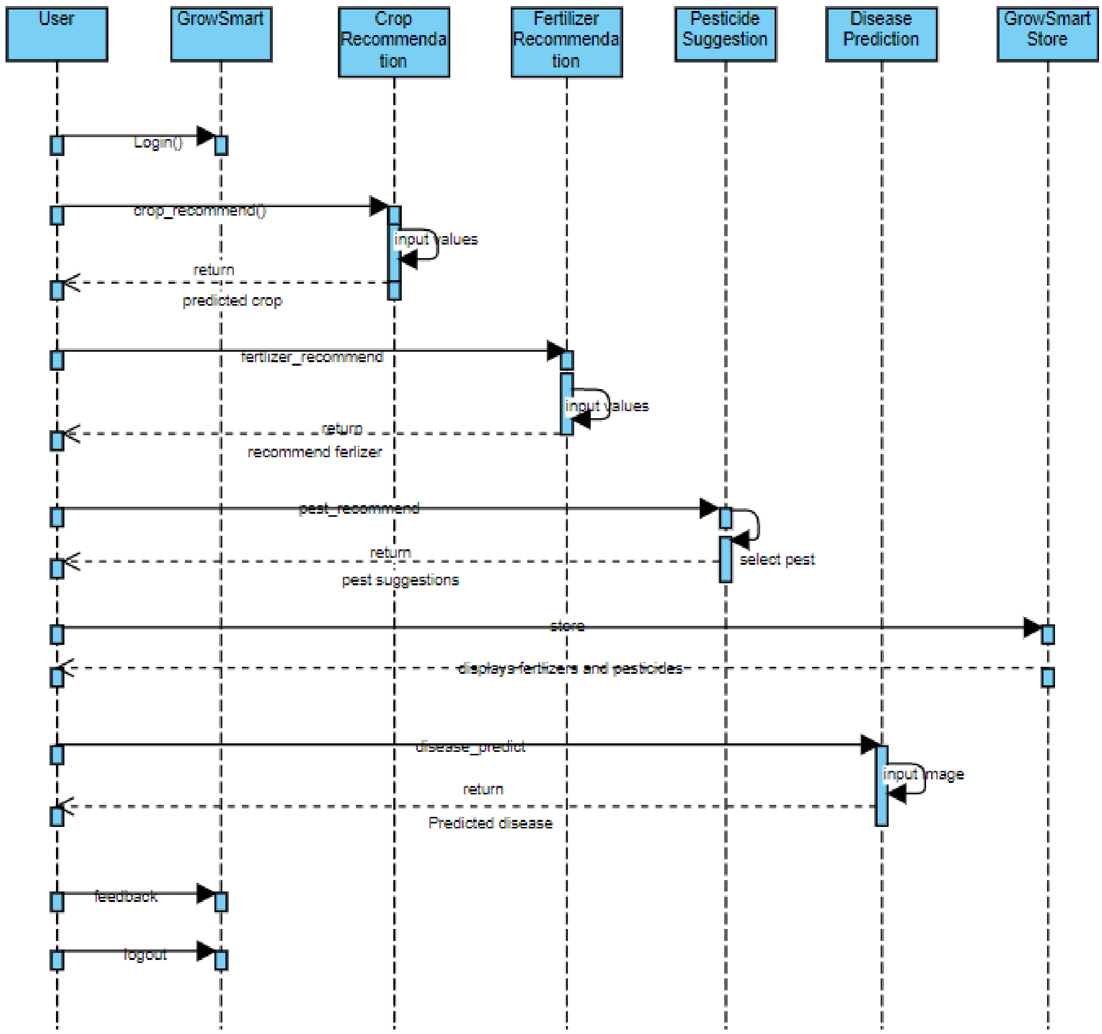


Fig 4.1.4: Sequence Diagram

4.2 Model Design

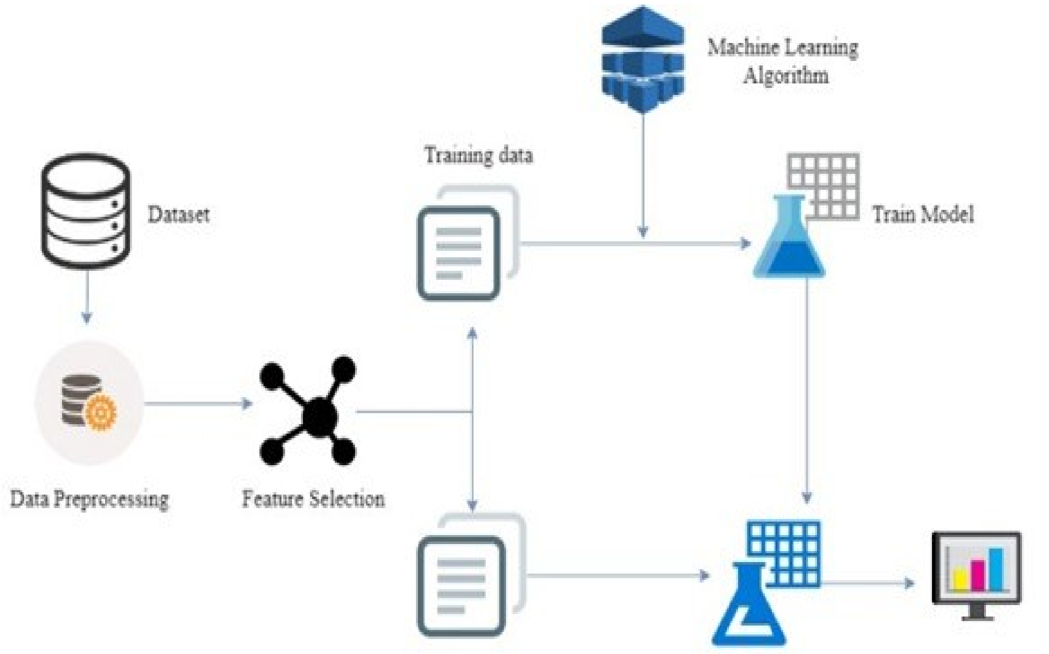


Fig 4.2: Model Design

4.3 Flow Diagram

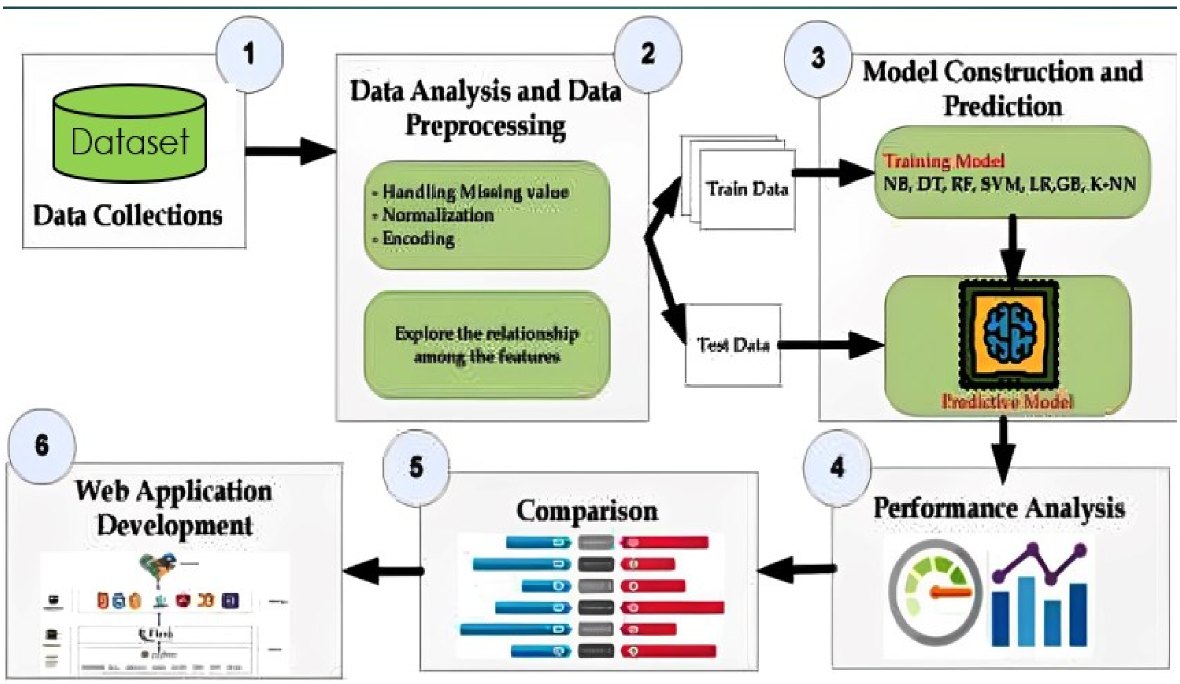


Fig 4.3: Flow Diagram

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 Software Environment

5.2 Machine Learning

Machine learning is the study of how to program computers to learn and behave like humans without explicit programming.

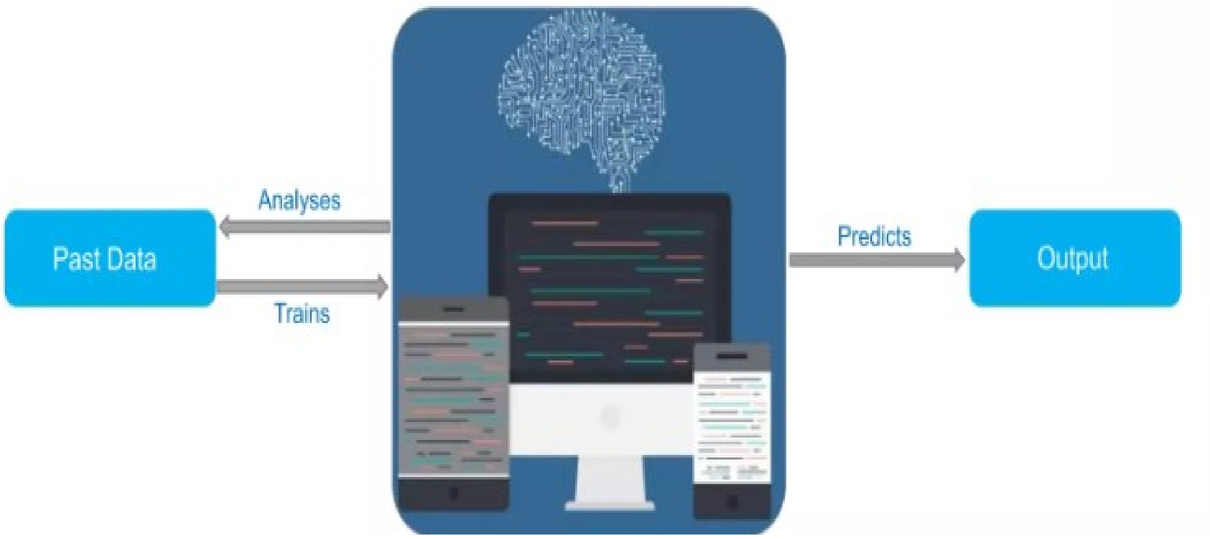


Fig 5.2: Machine Learning Representation

5.2.1 Processes Involved In Machine Learning

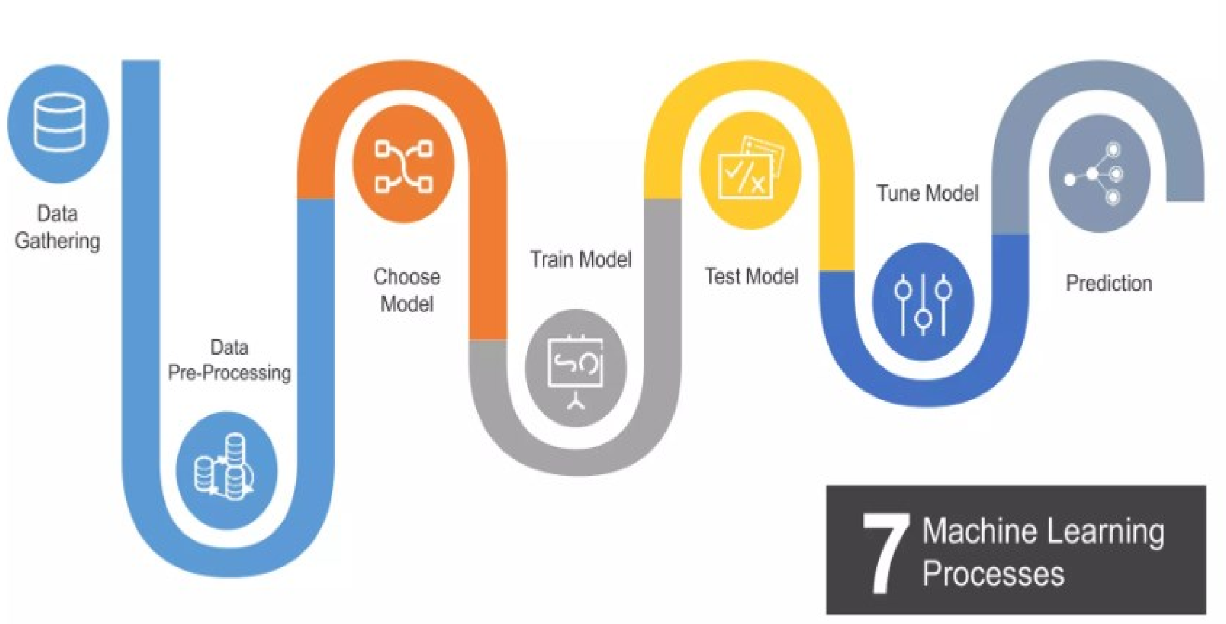


Fig 5.2.1: Processes in Machine Learning

5.3 Machine Learning Algorithms Used

5.3.1 Logistic Regression

Logistic Regression is a statistical method used to analyze data and make predictions about the likelihood of an event occurring. It is a type of regression analysis that is used when the dependent variable is binary or categorical in nature. Figuring out how the dependent variable and one or more independent factors are related is the aim of logistic regression. It uses a logistic function to transform the input values into a probability value between 0 and 1. The logistic regression model estimates the coefficients of the independent variables that maximize the likelihood of the observed data. These coefficients are used to calculate the probability of the dependent variable taking a certain value.

5.3.2 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a non-parametric algorithm used for classification and regression tasks in machine learning. It works by finding the K-nearest points to a given data point based on a chosen distance metric. The class of the data point is then determined by the majority class of its K-nearest neighbours in the training set. Regression uses the K-nearest neighbours to calculate the predicted number. It is simple to implement and works well in low-dimensional spaces. However, it can be computationally expensive in high-dimensional spaces and requires enough labelled data. The choice of K affects the accuracy of the model, with small values leading to more flexible models and larger values leading to more rigid models. KNN can also suffer from the curse of dimensionality, where the data becomes increasingly sparse, and the distance metric becomes less meaningful in high-dimensional spaces.

5.3.3 Support Vector Machine

A machine learning method known as a support vector machine is employed for regression and classification analysis. Finding the hyperplane that best divides data into various groups is how it functions. The hyperplane is selected to maximize the margin, which is the separation between the hyperplane and the closest data values for each class. By utilizing kernel functions, this algorithm is able to manage data that can be separated in both linear and non-linear ways. The use of regularization parameters allows them to effectively operate in high-dimensional environments and avoid overfitting.

5.3.4 Naïve Bayes

A probabilistic method used in machine learning for classification jobs is called naive Bayes. It is founded on the Bayes theorem, which states that the likelihood of the evidence given the hypothesis, multiplied by the prior probability of the hypothesis, determines the probability of the hypothesis given some observed evidence. The "naive" premise behind Naive Bayes is that, given the class label, the features are conditionally independent. This makes the algorithm more computationally effective and simplifies the likelihood estimate.

5.3.5 Decision Tree

For classification and regression analysis, a decision tree is a form of machine learning algorithm. A test on a feature is represented by each internal node, the test's result is represented by each branch, and a class name or a numerical value is represented by each leaf node in this tree-like structure. Because they are simple to understand and can manage both categorical and numerical data, decision trees are frequently used in business and industrial applications.

5.3.6 Random Forest

Random Forest is a machine learning algorithm that constructs with multiple decision trees and takes one final prediction. Each decision tree is grown using a random subset of features and data samples, making them less prone to overfitting. The final prediction of the Random Forest is the majority vote or average prediction of all the individual trees. This is most frequently used in classification and regression tasks. This algorithm can handle both categorical and continuous data, and also handles missing values. It can identify feature importance, making it useful for feature selection and data visualization. Random Forest is more efficient and can handle huge datasets.

5.3.7 Gradient Boosting

Gradient Boosting is a machine learning algorithm which is used for both regression and classification problems. It involves creating an ensemble of weak models like decision trees that are trained sequentially to correct the errors of the previous models. Each subsequent model tries to minimize the residual errors of the previous models using gradient descent optimization. The final prediction is a weighted sum of the predictions of all the weak models in the ensemble. The learning rate, which controls the contribution of each individual model to the final prediction, is an important hyperparameter in this algorithm. It is known for its high accuracy and ability to handle complex datasets.

5.4 NumPy

NumPy is the library for the python language which provide arrays and matrices for the storing the large data. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. It provides and support high-level and more mathematical functions which works on these NumPy arrays and matrices.

5.5 Pandas

Pandas is the library for the python language which is used for data manipulation and data science/data analysis. It is built on top of another package NumPy, which provides support for multi-dimensional arrays. It provides some data structures and operations for the manipulation and analysis.

5.6 Sklearn

Scikit-learn or sklearn is probably the most useful library for machine learning in Python. It is used for implementing the different machine learning algorithms on the given data set. It provides the different algorithms and models to implement easily and accurately

5.7 Installing Anaconda Python, Jupyter Notebook

- Navigate to Anaconda distribution website and find the installer.
- It's recommended that you install the Anaconda distribution to get Jupyter notebook; this distribution contains some useful packages and an environment manager to keep your packages installed and up to date.
- Select Download from the main menu, and then click on the Download.
- This will take you to a screen where you select your operating system for the installation.

5.7.1 Steps to Launch Jupyter Notebook

- You are done with the successful installation of Python and Anaconda in our system, Now let's set up Jupyter Notebook.
- To launch Jupyter Notebook via command line, simply open our Anaconda Windows Command Prompt. Here, type and run Jupiter Notebook as shown in figure 5.7.1

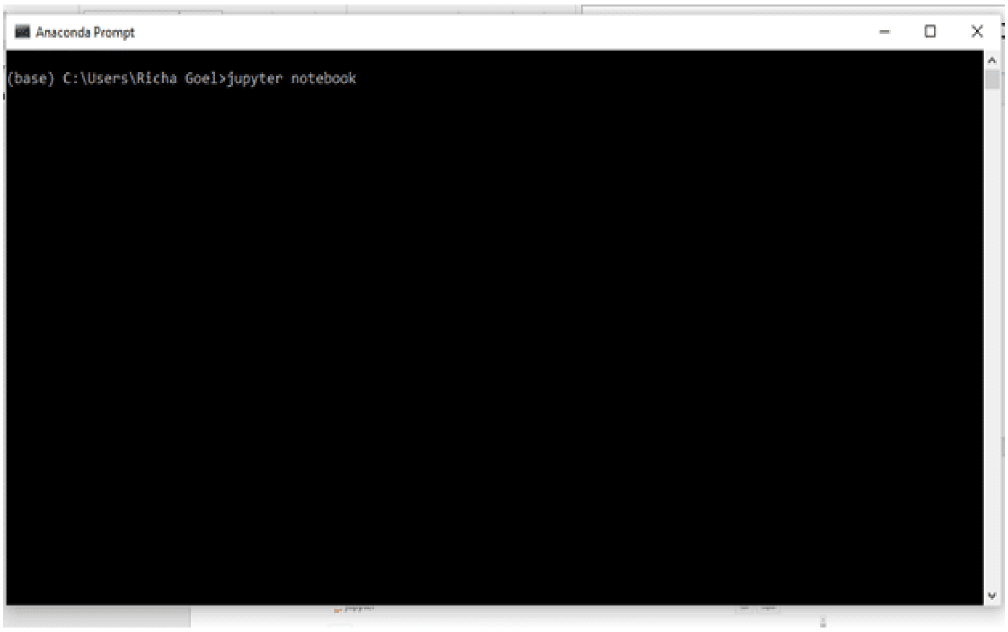


Fig 5.7.1.1: Command Prompt

A Jupyter Notebook dashboard will open on our default Your browser as shown in figure 5.7.2

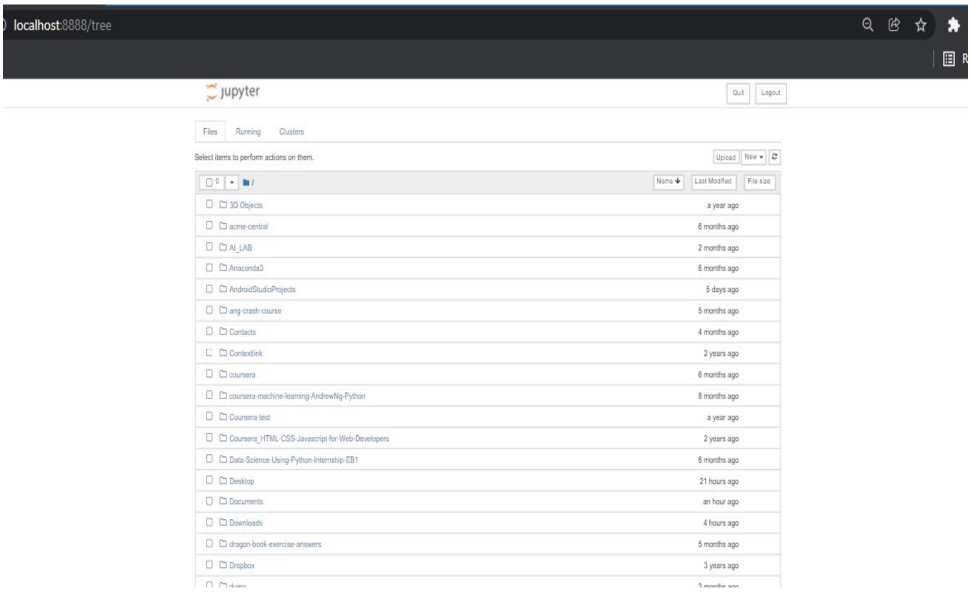


Fig 5.7.1.2: Jupyter-Notebook Home Page

5.8 Source Code:

Installing and importing the required packages:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('crop_recommendation.csv')
df.head(10)
```

	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

```
df.tail(10)
```

	N	P	K	temperature	humidity	ph	rainfall	label
2190	103	40	30	27.309018	55.196224	6.348316	141.483164	coffee
2191	118	31	34	27.548230	62.881792	6.123796	181.417081	coffee
2192	106	21	35	25.627355	57.041511	7.428524	188.550654	coffee
2193	116	38	34	23.292503	50.045570	6.020947	183.468585	coffee
2194	97	35	26	24.914610	53.741447	6.334610	166.254931	coffee
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

df.shape

(2200, 8)

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   N            2200 non-null   int64  
 1   P            2200 non-null   int64  
 2   K            2200 non-null   int64  
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object 
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

df.describe()

	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

df.isnull().sum()

```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

df['label']. unique()

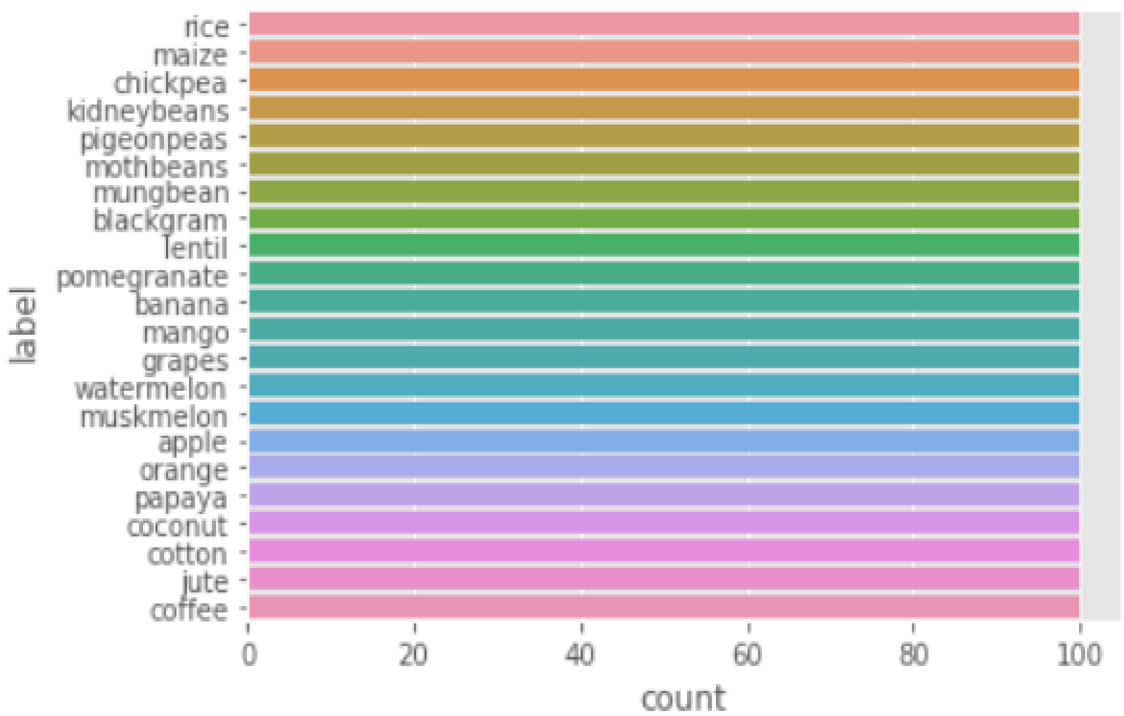
```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
      dtype=object)
```

df['label'].value_counts()

```
rice           100
maize          100
jute           100
cotton         100
coconut        100
papaya         100
orange         100
apple          100
muskmelon      100
watermelon     100
grapes         100
mango          100
banana         100
pomegranate    100
lentil         100
blackgram      100
mungbean       100
mothbeans      100
pigeonpeas     100
kidneybeans    100
chickpea       100
coffee         100
Name: label, dtype: int64
```

sns.countplot(y = 'label',data = df)

```
<AxesSubplot:xlabel='count', ylabel='label'>
```



X = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]

Y = df['label']

X

	N	P	K	temperature	humidity	ph	rainfall
0	90	42	43	20.879744	82.002744	6.502985	202.935536
1	85	58	41	21.770462	80.319644	7.038096	226.655537
2	60	55	44	23.004459	82.320763	7.840207	263.964248
3	74	35	40	26.491096	80.158363	6.980401	242.864034
4	78	42	42	20.130175	81.604873	7.628473	262.717340
...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507
2196	99	15	27	27.417112	56.636362	6.086922	127.924610
2197	118	33	30	24.131797	67.225123	6.362608	173.322839
2198	117	32	34	26.272418	52.127394	6.758793	127.175293
2199	104	18	30	23.603016	60.396475	6.779833	140.937041

2200 rows × 7 columns

5.8.1 Test and Train Split

We split the dataset in the ratio 80:20, where 80 is for the training and 20 is for testing.

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=0)

# Checking dimensions
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("Y_train shape:", Y_train.shape)
print("Y_test shape:", Y_test.shape)
```

```
X_train shape: (1760, 7)
X_test shape: (440, 7)
Y_train shape: (1760,)
Y_test shape: (440,)
```

5.8.2 Comparative Analysis

Using 7 different algorithms to find which algorithm performs well.

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state = 42)
logreg.fit(X_train, Y_train)

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 24, metric = 'minkowski', p = 2)
knn.fit(X_train, Y_train)

from sklearn.svm import SVC
#svc = SVC(kernel = 'linear', random_state = 42)
svc = SVC()
svc.fit(X_train, Y_train)

from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train, Y_train)

from sklearn.tree import DecisionTreeClassifier
dectree = DecisionTreeClassifier(criterion = 'entropy', random_state = 42)
dectree.fit(X_train, Y_train)

from sklearn.ensemble import RandomForestClassifier
ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy', random_state = 42)
ranfor.fit(X_train, Y_train)

from sklearn.ensemble import GradientBoostingClassifier
gbc=GradientBoostingClassifier()
gbc.fit(X_train,Y_train)
```

5.8.3 Accuracy Of ML Algorithms - Training Accuracy

accuracy["Lin R"]=linreg.score(X_train, Y_train)

accuracy["Log R"]=logreg.score(X_train, Y_train)

accuracy["knn"]=knn.score(X_train, Y_train)

accuracy["svc"]=svc.score(X_train, Y_train)

accuracy["nb"]=nb.score(X_train, Y_train)

accuracy["dectree"]=dectree.score(X_train, Y_train)

accuracy["ranfor"]=ranfor.score(X_train, Y_train)

accuracy["gbc"]=gbc.score(X_train, Y_train)

```
accuracy
```

```
{'Log R': 0.9693181818181819,
 'knn': 0.9670454545454545,
 'svc': 0.9767045454545454,
 'nb': 0.9954545454545455,
 'dectree': 1.0,
 'ranfor': 0.9994318181818181,
 'gbc': 1.0}
```

```
import matplotlib.pyplot as plt
algo=list(accuracy.keys())
accu=list(accuracy.values())
plt.bar(range(len(accuracy)),accu,tick_label=algo)
```

```
<BarContainer object of 7 artists>
```

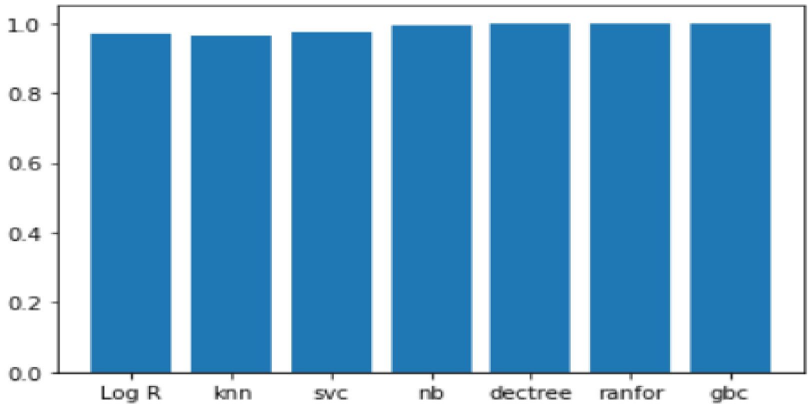


Fig 5.8.3: Training accuracy

5.8.4 Accuracy Of ML Algorithms - Testing Accuracy

Y_pred_linreg = linreg.predict(X_test)

Y_pred_logreg = logreg.predict(X_test)

Y_pred_knn = knn.predict(X_test)

Y_pred_svc = svc.predict(X_test)

Y_pred_nb = nb.predict(X_test)

Y_pred_dectree = dectree.predict(X_test)

Y_pred_ranfor = ranfor.predict(X_test)

Y_pred_gbc = gbc.predict(X_test)

from sklearn.metrics import accuracy_score

accuracy_linreg = r2_score(Y_test, Y_pred_linreg)

accuracy_logreg = accuracy_score(Y_test, Y_pred_logreg)

accuracy_knn = accuracy_score(Y_test, Y_pred_knn)

accuracy_svc = accuracy_score(Y_test, Y_pred_svc)

accuracy_nb = accuracy_score(Y_test, Y_pred_nb)

accuracy_dectree = accuracy_score(Y_test, Y_pred_dectree)

accuracy_ranfor = accuracy_score(Y_test, Y_pred_ranfor)

accuracy_gbc = accuracy_score(Y_test, Y_pred_gbc)

```
accuracy1
```

```
{'Log R': 0.9681818181818181,
 'knn': 0.9613636363636363,
 'svc': 0.9772727272727273,
 'nb': 0.9931818181818182,
 'dectree': 0.9931818181818182,
 'ranfor': 0.9954545454545455,
 'GB': 0.9954545454545455}
```

```
import matplotlib.pyplot as plt
algo=list(accuracy1.keys())
accu=list(accuracy1.values())
plt.bar(range(len(accuracy1)),accu,tick_label=algo)
```

```
<BarContainer object of 7 artists>
```

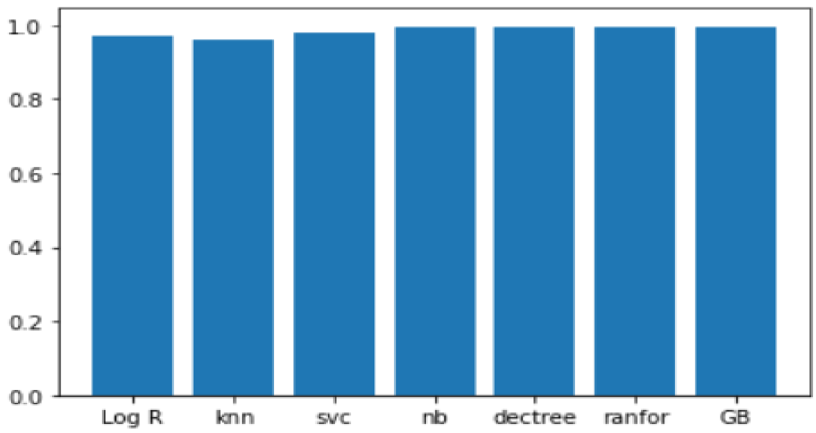


Fig 5.8.4: Testing Accuracy

5.8.5 Test and Train Accuracy Plot

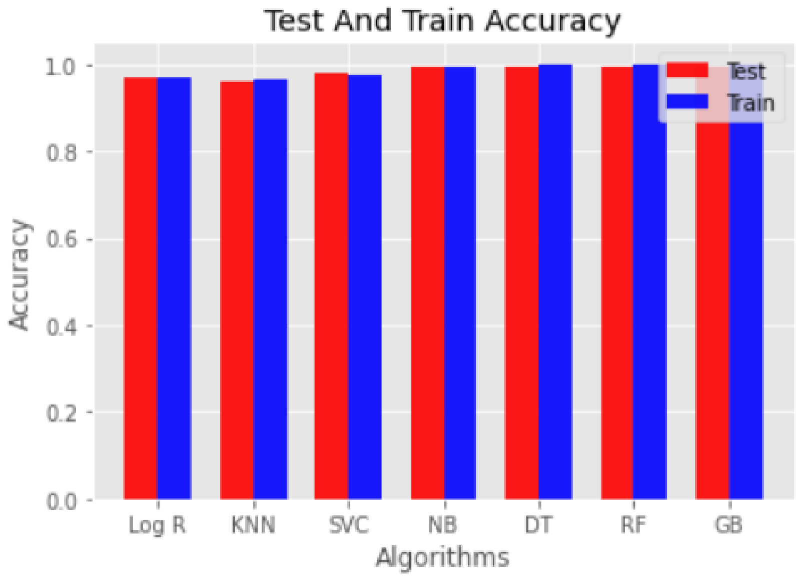


Fig 5.8.5: Test Train plot

5.8.6 Finding Maximum Accuracy Algorithm

```
m=max(list(accuracy1.values()))
print("The Algorithm which gives maximum accuracy...")
for i in accuracy1:
    if accuracy1[i]==m:
        print("{} : {}".format(i,accuracy1[i]*100))
```

O/P: The Algorithm which gives maximum accuracy...

ranfor : 99.54545454545455

GB : 99.54545454545455

We have found that two algorithms are performing well. So after testing with new data we have concluded that **"Random Forest"** is more accurate than Gradient Boosting Algorithm.

Now we are going to use Random Forest model for further process.

5.8.7 Saving Random Forest Model

```
import pickle
RF_pkl_filename = 'RF.pkl'
RF_Model_pkl = open(RF_pkl_filename, 'wb')
```

```
pickle.dump(RF, RF_Model_pkl)

RF_Model_pkl.close()
```

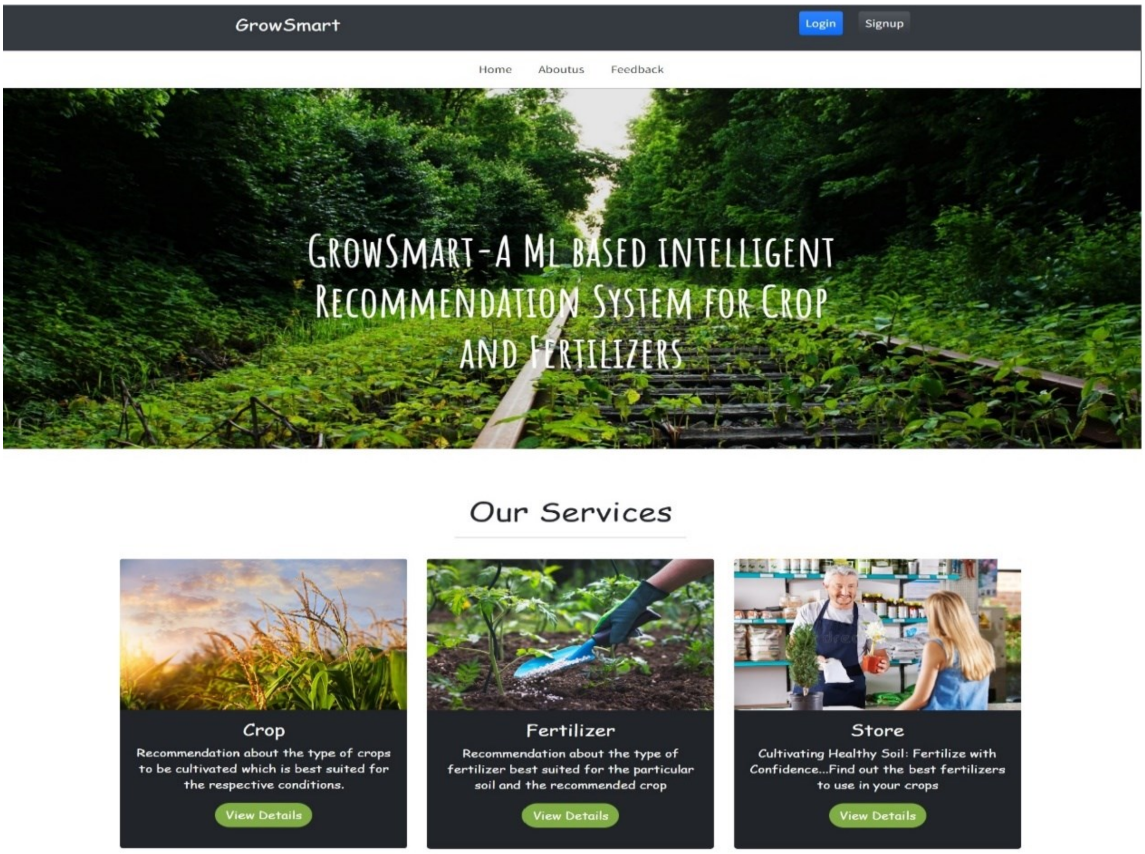


Fig 5.8.7: GrowSmart Web Application

5.9 Input & Output

5.9.1 Crop Recommendation System

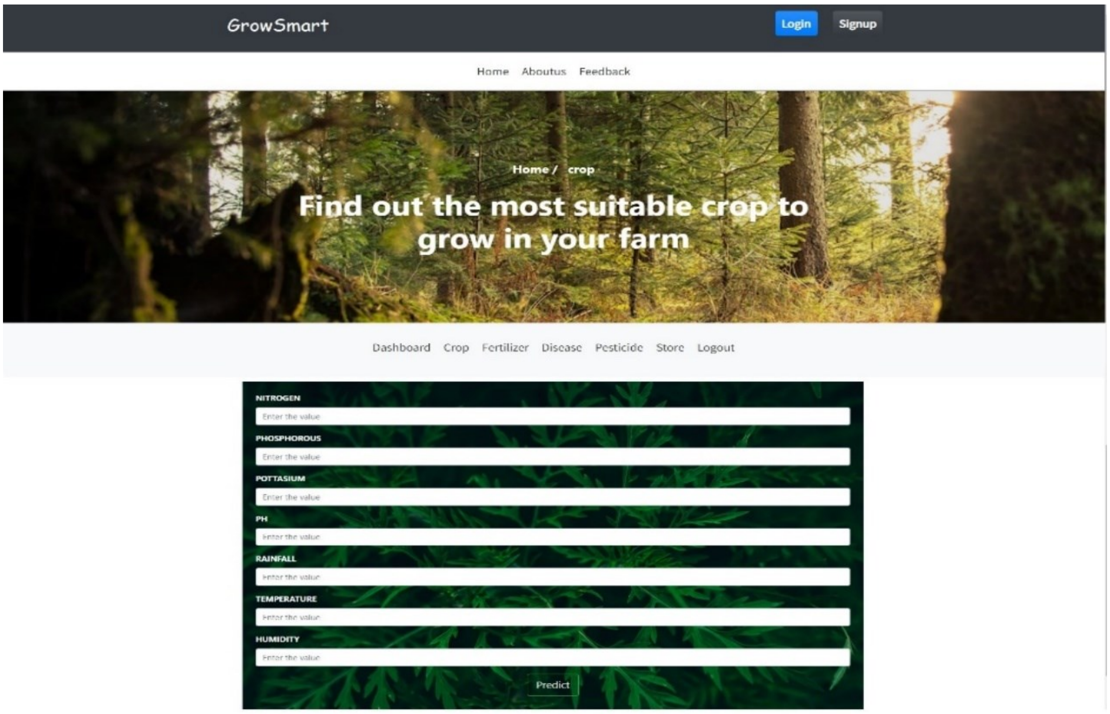


Fig 5.9.1.1: Crop Recommendation System

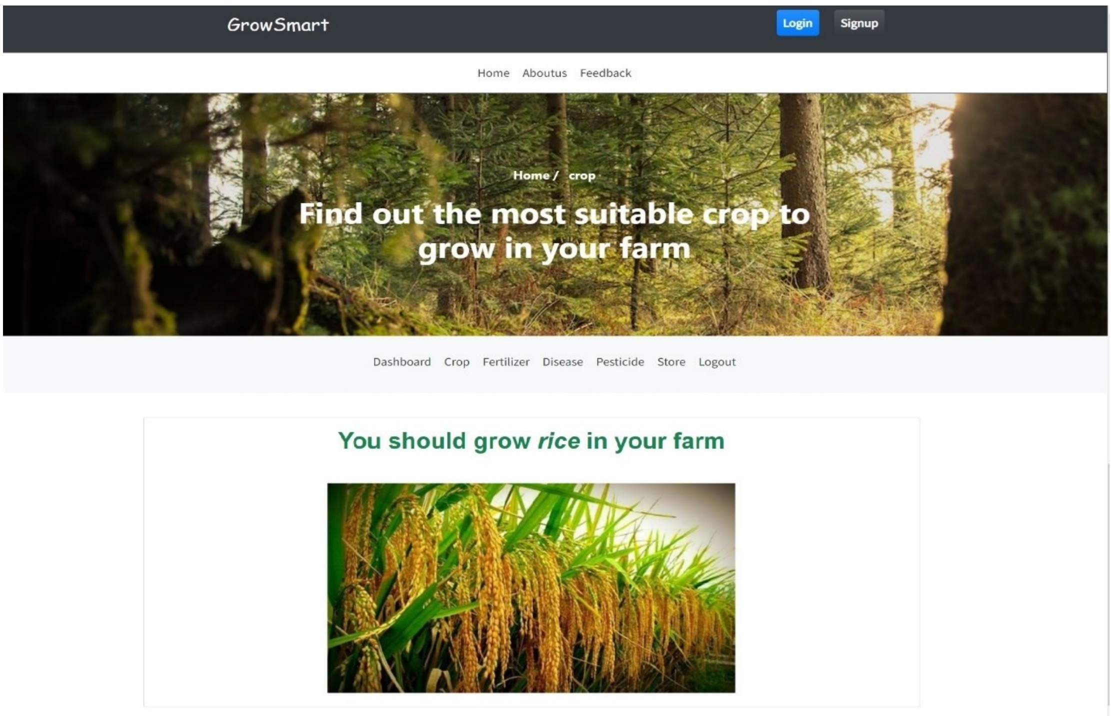


Fig 5.9.1.2: Crop Recommendation System Result

5.9.2 Fertilizer Recommendation System

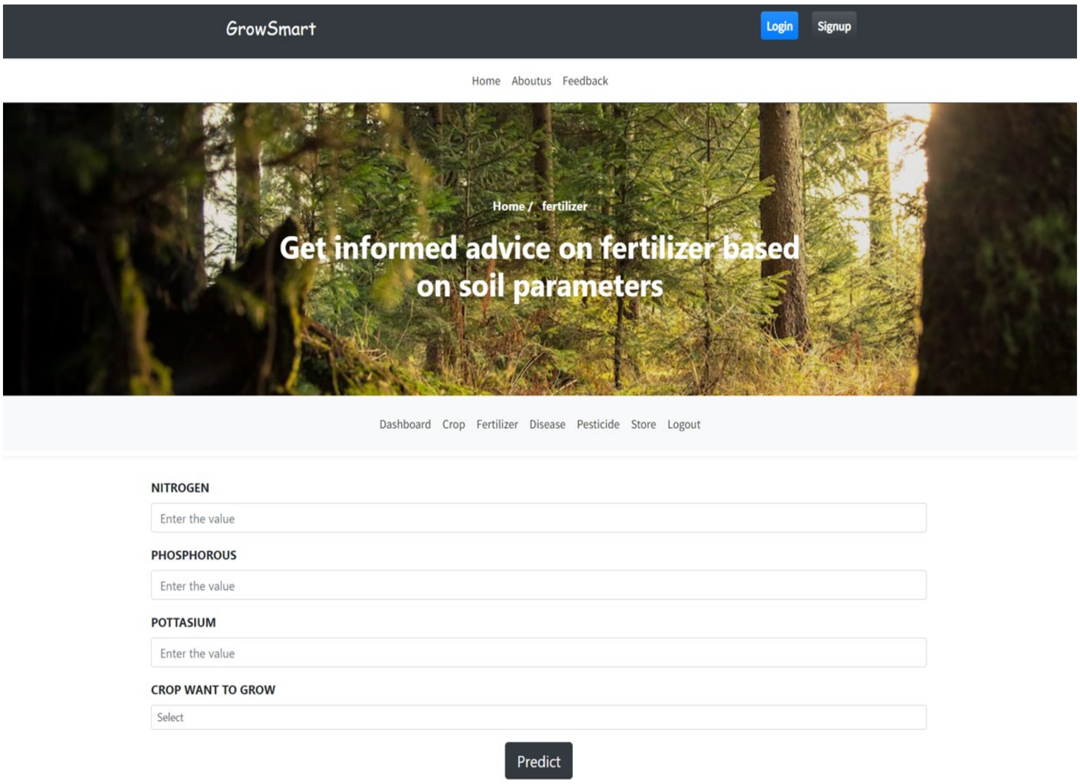
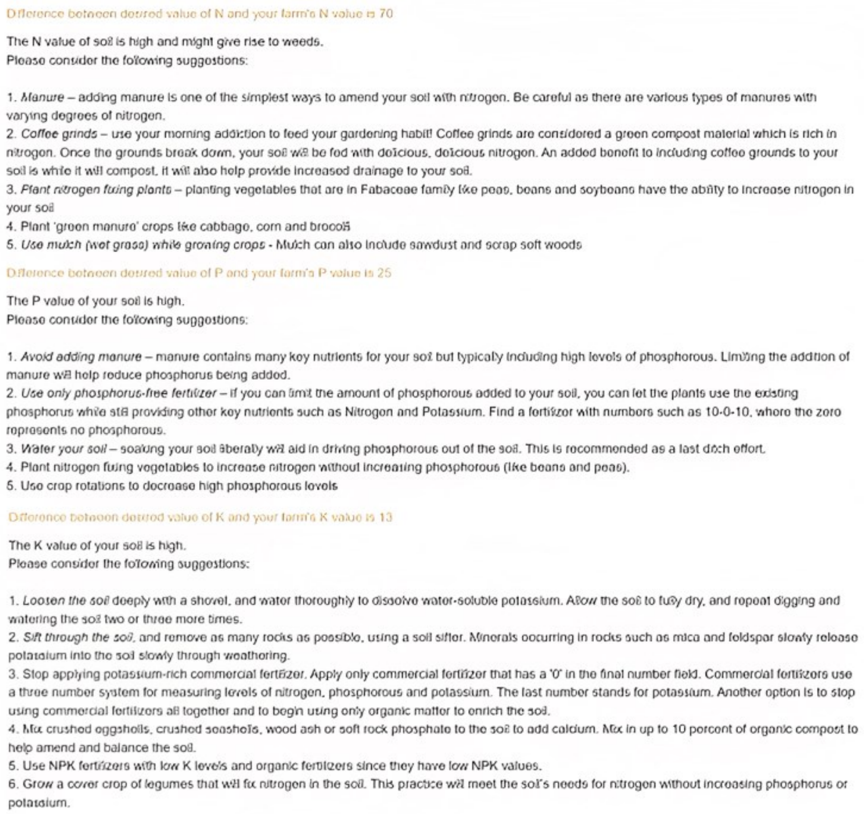


Fig 5.9.2.1: Fertilizer Recommendation System



Difference between desired value of N and your farm's N value is 70

The N value of soil is high and might give rise to weeds.
Please consider the following suggestions:

1. *Manure* – adding manure is one of the simplest ways to amend your soil with nitrogen. Be careful as there are various types of manures with varying degrees of nitrogen.
2. *Coffee grinds* – use your morning addiction to feed your gardening habit! Coffee grinds are considered a green compost material which is rich in nitrogen. Once the grounds break down, your soil will be fed with delicious, delicious nitrogen. An added benefit to including coffee grounds to your soil is while it will compost, it will also help provide increased drainage to your soil.
3. *Plant nitrogen fixing plants* – planting vegetables that are in Fabaceae family like peas, beans and soybeans have the ability to increase nitrogen in your soil
4. Plant 'green manure' crops like cabbage, corn and brocolli
5. *Use mulch (wet grass) while growing crops* - Mulch can also include sawdust and scrap soft woods

Difference between desired value of P and your farm's P value is 25

The P value of your soil is high.
Please consider the following suggestions:

1. *Avoid adding manure* – manure contains many key nutrients for your soil but typically including high levels of phosphorous. Limiting the addition of manure will help reduce phosphorus being added.
2. *Use only phosphorus-free fertilizer* – if you can limit the amount of phosphorous added to your soil, you can let the plants use the existing phosphorus while still providing other key nutrients such as Nitrogen and Potassium. Find a fertilizer with numbers such as 10-0-10, where the zero represents no phosphorous.
3. *Water your soil* – soaking your soil liberally will aid in driving phosphorous out of the soil. This is recommended as a last ditch effort.
4. Plant nitrogen fixing vegetables to increase nitrogen without increasing phosphorous (like beans and peas).
5. Use crop rotations to decrease high phosphorous levels

Difference between desired value of K and your farm's K value is 13

The K value of your soil is high.
Please consider the following suggestions:

1. *Loosen the soil* deeply with a shovel, and water thoroughly to dissolve water-soluble potassium. Allow the soil to fully dry, and repeat digging and watering the soil two or three more times.
2. *Sift through the soil*, and remove as many rocks as possible, using a soil sifter. Minerals occurring in rocks such as mica and feldspar slowly release potassium into the soil slowly through weathering.
3. Stop applying potassium-rich commercial fertilizer. Apply only commercial fertilizer that has a '0' in the final number field. Commercial fertilizers use a three number system for measuring levels of nitrogen, phosphorous and potassium. The last number stands for potassium. Another option is to stop using commercial fertilizers all together and to begin using only organic matter to enrich the soil.
4. Mix crushed eggshells, crushed seashells, wood ash or soft rock phosphate to the soil to add calcium. Mix in up to 10 percent of organic compost to help amend and balance the soil.
5. Use NPK fertilizers with low K levels and organic fertilizers since they have low NPK values.
6. Grow a cover crop of legumes that will fix nitrogen in the soil. This practice will meet the soil's needs for nitrogen without increasing phosphorus or potassium.

Fig 5.9.2.2: Fertilizer Recommendation System Result

5.9.3 Pesticide Recommendations System

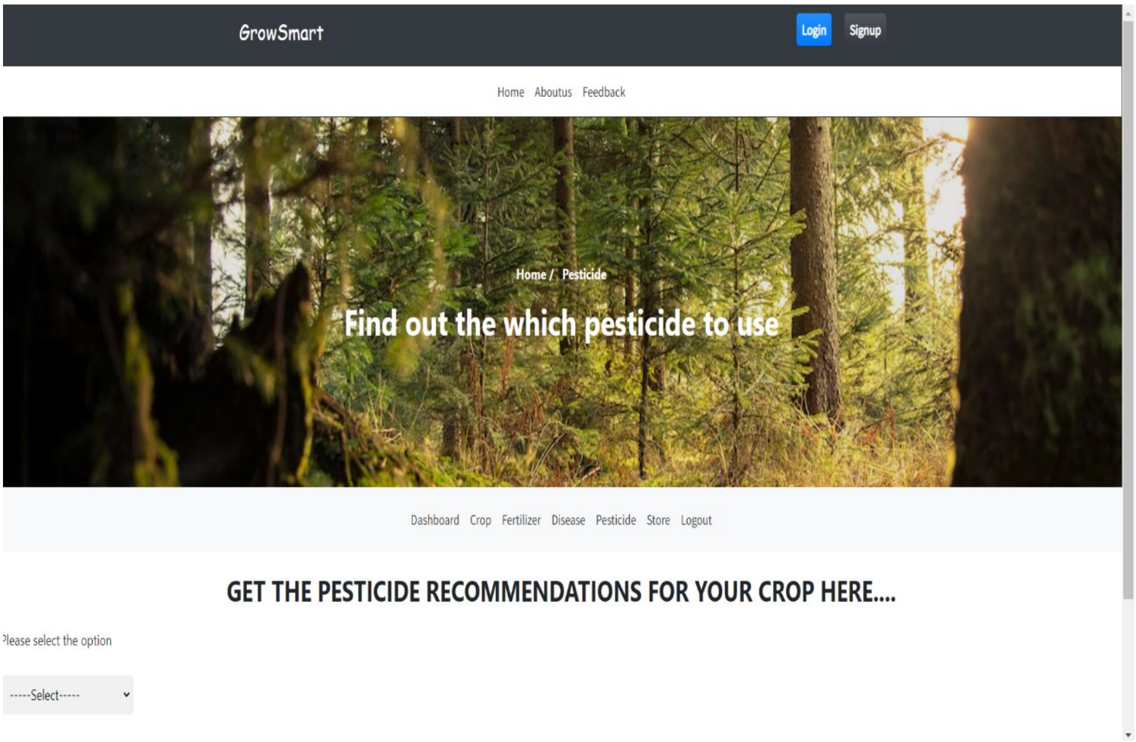


Fig 5.9.3.1: Pesticide Recommendation System

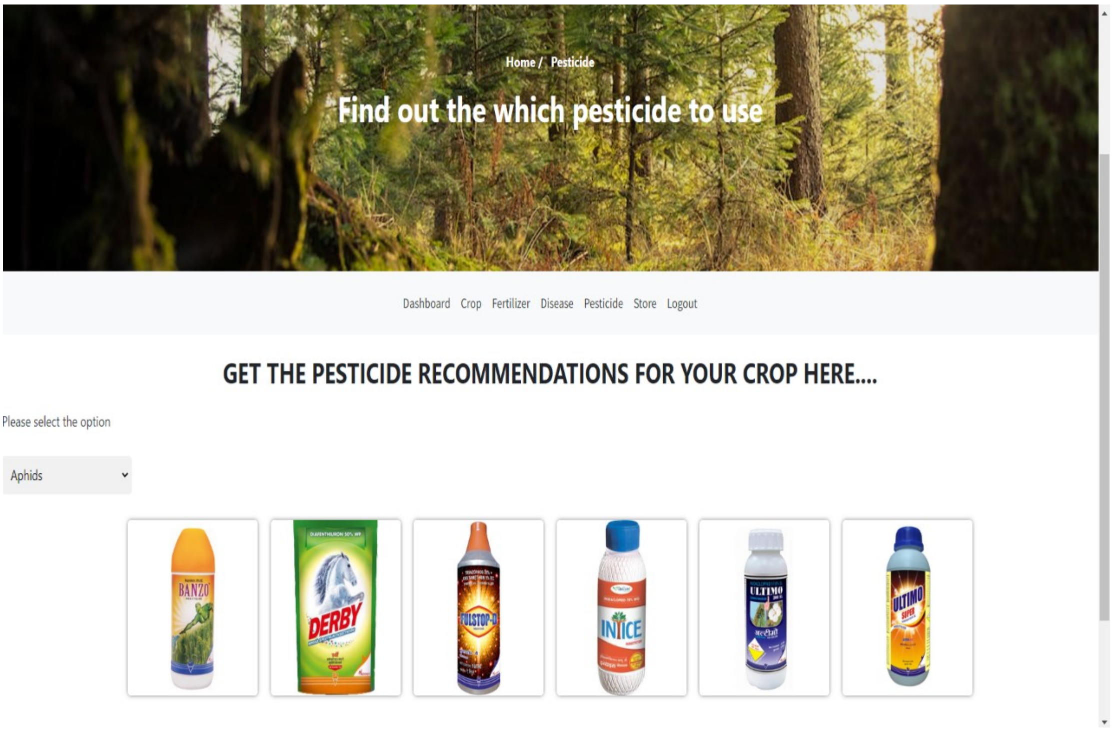


Fig 5.9.3.2: Pesticide Recommendation System Result

5.9.4 Disease Prediction System

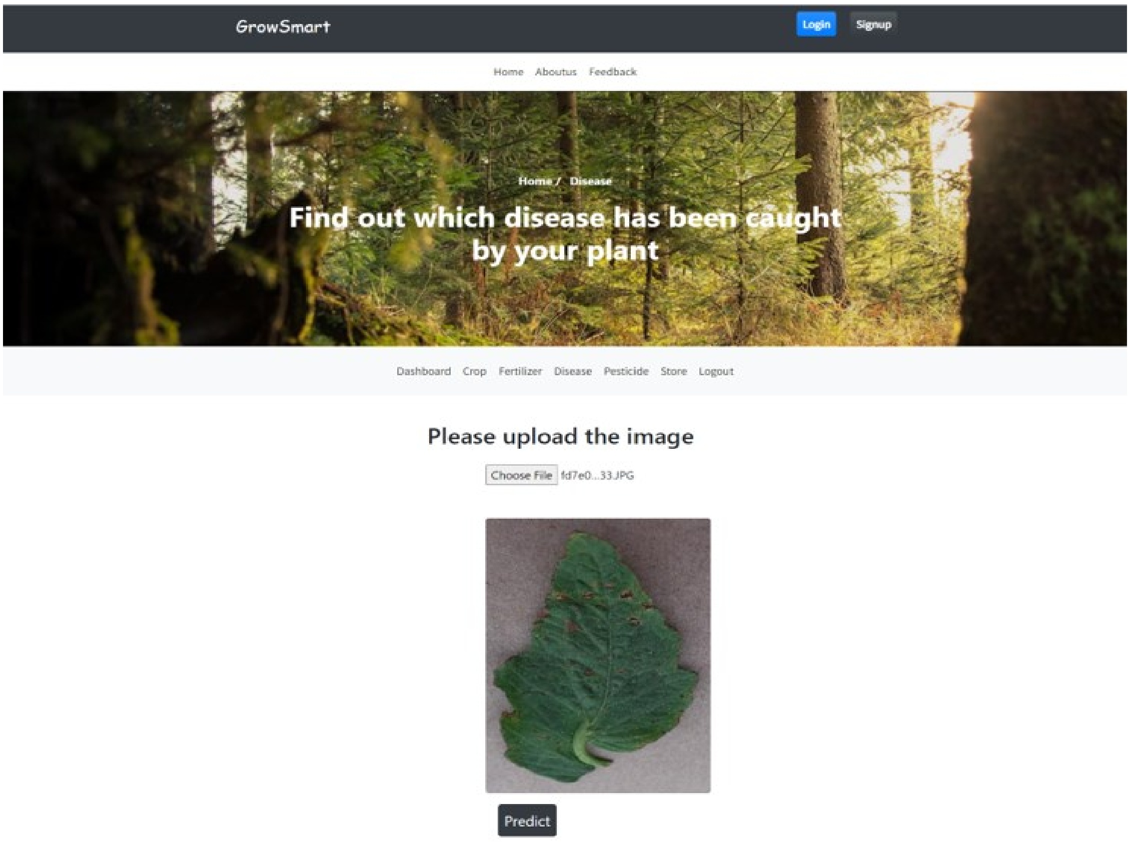
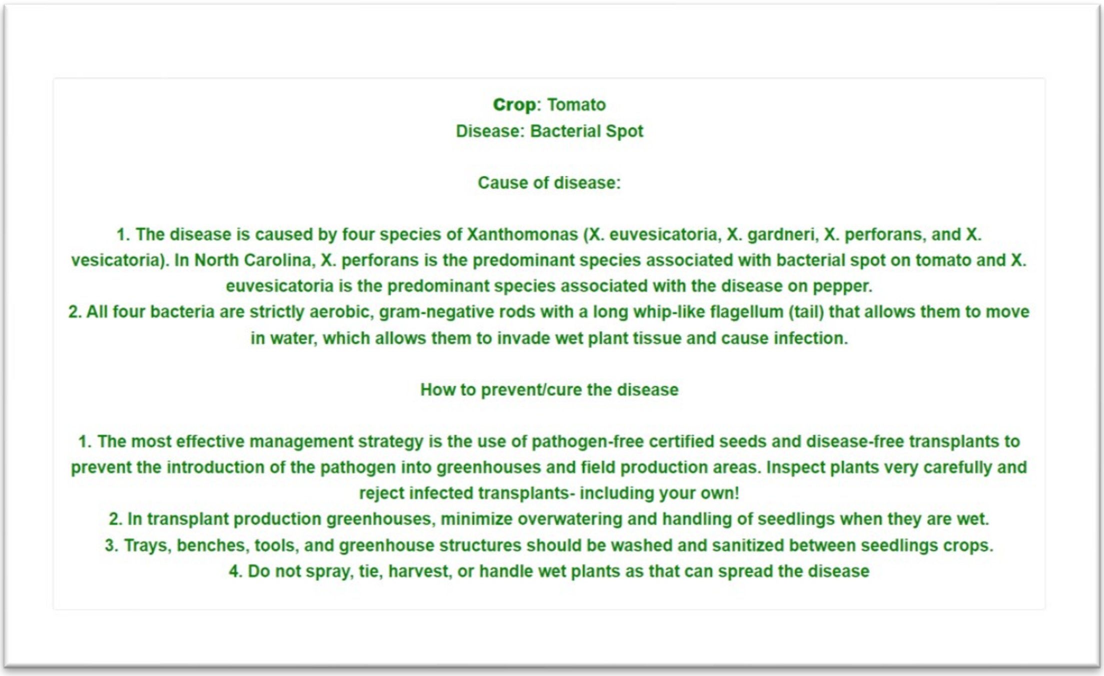


Fig 5.9.4.1: Disease Prediction System



Crop: Tomato
Disease: Bacterial Spot

Cause of disease:

1. The disease is caused by four species of Xanthomonas (X. euvesicatoria, X. gardneri, X. perforans, and X. vesicatoria). In North Carolina, X. perforans is the predominant species associated with bacterial spot on tomato and X. euvesicatoria is the predominant species associated with the disease on pepper.
2. All four bacteria are strictly aerobic, gram-negative rods with a long whip-like flagellum (tail) that allows them to move in water, which allows them to invade wet plant tissue and cause infection.

How to prevent/cure the disease

1. The most effective management strategy is the use of pathogen-free certified seeds and disease-free transplants to prevent the introduction of the pathogen into greenhouses and field production areas. Inspect plants very carefully and reject infected transplants- including your own!
2. In transplant production greenhouses, minimize overwatering and handling of seedlings when they are wet.
3. Trays, benches, tools, and greenhouse structures should be washed and sanitized between seedlings crops.
4. Do not spray, tie, harvest, or handle wet plants as that can spread the disease

Fig 5.9.4.2: Disease Prediction System Result

5.9.5 GrowSmart Fertilizer & Pesticide Store

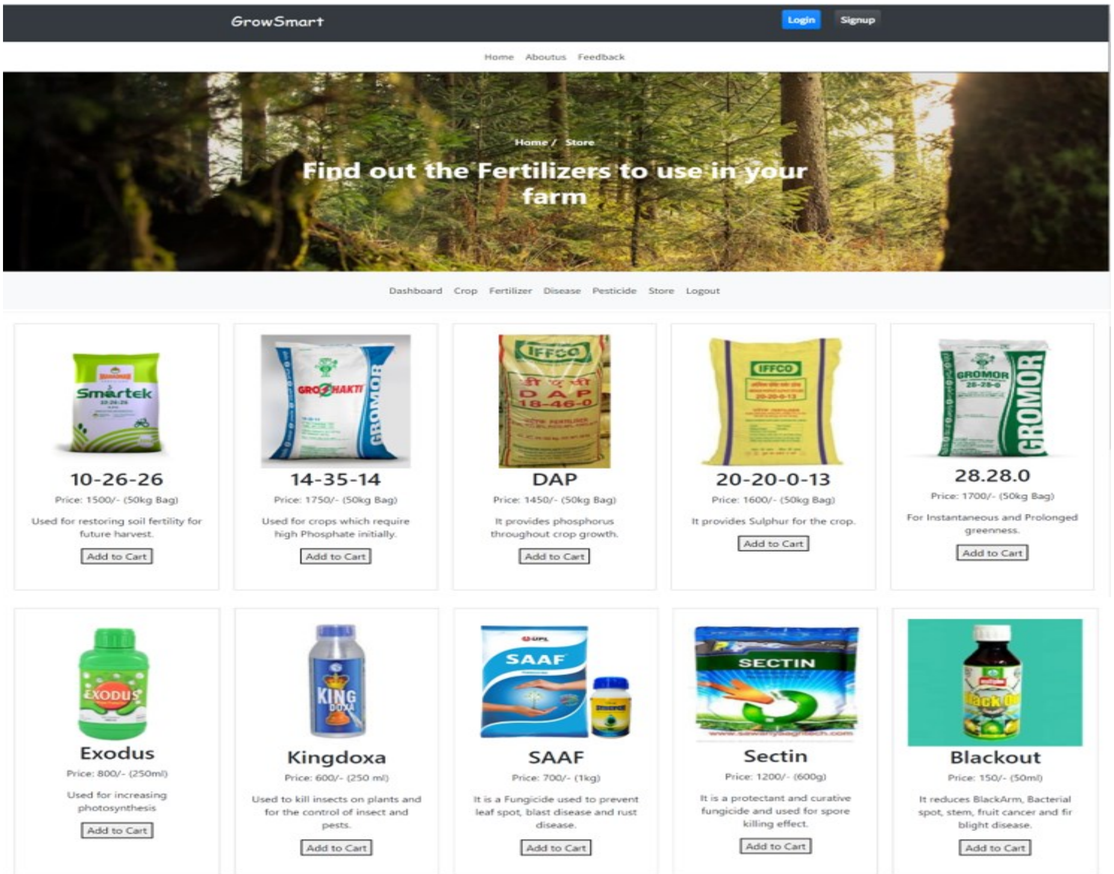


Fig 5.9.5: GrowSmart Store

CHAPTER 6
TESTING

6.1 Introduction:

Testing is the process of assessing a software application or system to see if it satisfies the requirements and performs as intended. Software is tested to find any flaws or problems and to make sure it is dependable, effective, and up to user expectations.

6.1.1 Testing Methods

There are several testing methods that can be used to evaluate software applications or systems. These include:

1. **Unit Testing:**

   Unit testing is the process of testing individual units or components of software applications to ensure that each unit works as expected.
2. **Integration Testing:**

   Integration testing method involves testing the integration of different software components to ensure that they work together seamlessly.
3. **System Testing:**

   System testing involves testing the entire software system to ensure that it meets the specified requirements and functions as expected.
4. **Acceptance Testing:**

   Acceptance testing method involves testing the software application or system to ensure that it meets the user's requirements and is ready for deployment.
5. **Performance Testing:**

   Performance testing involves testing the performance of the software application or system to ensure that it can handle a specific load and performs efficiently.
6. **Security Testing:**

   Security testing involves testing the software application or system to ensure that it is secure from potential threats and vulnerabilities.

6.2 Validation of Objectives:

S.NO.	OBJECTIVES	STATUS
1	To build a robust model to provide correct and accurate prediction of crop sustainability for the soil type and climatic conditions	Successful
2	To build user friendly and interactive application	Successful
3	Provide recommendation of the most effective suitable crops within the area in order that the farmer does not incur any losses	Successful
4	Provide fertilizer suggestion and recommendations based on the crop and N, P, and K values	Successful
5	Provide disease prediction and Pesticide recommendation System for crops	Successful
6	To identify the pest and suggest a specific pesticide that is offered in India in accordance with ISO criteria	Successful
7	To build a best fertilizer and pesticide store	Successful

Table 3: Validation of Objectives

6.3 Conclusion:

The testing process for a Machine Learning and Flask web application is most important step in ensuring the overall quality and functionality of the system. Through various testing techniques such as unit testing, integration testing, and end-to-end testing, developers can detect and fix potential bugs and issues before they make it into production.

Furthermore, testing can also help ensure that the ML models used in the application are performing optimally and effectively, and that they are providing accurate and reliable results. This is mostly important for ML applications, where the quality of the model's predictions can have significant implications.

Overall, a thorough testing process can help developers ensure that their ML and Flask application is robust, reliable, and able to meet the needs of users. By investing in testing early in the development process, developers can save time and resources in the long run by detecting and addressing issues before they become major problems.

CHAPTER 7
CONCLUSION

7.1 Conclusion

Farmers in India are working arduously. They contribute to the nation's almost 1.4 billion people being fed. However, some natural elements that have the potential to destroy their crops and way of life put their productivity in jeopardy. GrowSmart would therefore help farmers to increase agricultural production, decrease soil erosion in cultivated fields, receive knowledgeable advice on organic fertilizers/other fertilizers, and identify the best crop by considering numerous qualities. This would offer a thorough prediction, making this idea advantageous for both farmers and the environment. Not only that, but another significant issue that would be resolved by this initiative is insect management.

7.2 Scope for Future Work

GrowSmart is not limited to current usage, it can be extended to many features as discussed below":

- GrowSmart currently limited with 22 crops. It can be improved by adding more crops. Moreover, in the future, fertilizers can also be added accordingly.
- The user must manually enter the values of temperature, humidity, and rainfall in Crop Recommendation. A weather API can be used by the administrator to retrieve real-time parameters by city and state.
- In Pesticide Recommendation, we have only provided only pesticide suggestions based on pest here we can improve it by detecting the pest using DL methods and provide recommendations based on the results. In order to take live photographs of pests and notify farmers about the pest, pesticide code can be combined with drone code. Farmers will receive this information by email or mobile.
- We will be incorporating more Machine Learning and Deep learning models for achieving better results.

REFERENCES

[1] Rajak, Rohit Kumar, et al. "Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique." International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 12, 2017, pp. 951-952. IRJET.

[2] Dighe, Deepti, et al. "Crop Recommendation System for Precision Agriculture." IRJET, vol. 05, no. 11, 2018, pp. 476-480. IRJET.

[3] Mokarrama, Miftahul Jannat, and Mohammad Shamsul Arefin. "RSF: A Recommendation System for Farmers." Region 10 Humanitarian Technology Conference, vol. 2, no. 17, 2017.

[4] Gandge, Yogesh, and Sandhya. "A study on various data mining techniques for crop yield prediction." IEEE Xplore, 2017. IEEE Xplore.

[5] Mishra, Shruti, et al. Use of data mining in crop yield prediction. 2018. ResearchGate.

[6] Wu, Xiaoping, et al. A Large-Scale Benchmark Dataset for Insect Pest Recognition. 2019. IEEE Xplore.

[7] Kasinathan, Thenmozhi, et al. Insect classification and detection in field crops using modern machine learning techniques. 2020. Science Direct.

[8] Ding, Weiguang, and Graham Taylor. Automatic moth detection from trap images for pest management. 2016. Science Direct.

[9] TÜRKOĞLU, Muammer, and Davut HANBAY. Plant disease and pest detection using deep learning-based features. 2018. Turkish Journal of Electrical Engineering & Computer Sciences.

[10] Selvaraj, Michael Gomez, et al. AI-powered banana diseases and pest detection. 2019. Plant Methods.

[11] Dataset: https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset.

PUBLICATIONS

Dr.S.Kusuma, R M Sai Mohit, M Madhusudhan, B.Jagadeesh, K V Hariprasad, GrowSmart-An ML Based Intelligent Recommendation System For Crops And Fertilizers, NOVYI MIR research journal ISSN NO: 0130-7673 VOLUME 8 ISSUE 5 2023.

PUBLICATION PAPER:

NOVYI MIR Research Journal ISSN NO: 0130-7673

GrowSmart: An IoT and ML based Intelligent Recommendation System for Crops and Fertilizers

Dr.S.Kusuma
Assistant Professor
Dept.of CSE
MITS,Madanapalle
Andhra Pradesh

R M Sai Mohit
Dept.of CSE
MITS,Madanapalle
Andhra Pradesh

M Madhusudhan
Dept.of CSE
MITS,Madanapalle
Andhra Pradesh

B.Jagadeesh
Dept.of CSE
MITS,Madanapalle
Andhra Pradesh

K V Hariprasad
Dept.of CSE
MITS,Madanapalle
Andhra Pradesh

***Abstract*—Since agriculture is the foundation of a thriving economy like India, there is a great need to maintain agricultural sustainability. This field of study is expanding. As a result, it makes a considerable contribution to the economic and agricultural prosperity of nations all over the world. A nation's ability to produce food safely and securely depends on the efficient use of its agricultural land. Agriculture relies heavily on soil and environmental factors, such as temperature, humidity, and rainfall, to anticipate crops. In the past, farmers had control over the selection of the crop to be grown, the monitoring of its development, and the timing of its harvest. Now, however, the farming community finds it challenging to carry on due to the rapid changes in the environment. As a result, machine learning approaches have increasingly replaced traditional prediction methods. This work has used a number of these methods to calculate agricultural yield. The most frequent issue Indian farmers have is that they choose the wrong crop and don't utilise the right fertiliser for their soil. This will result in a major decrease in their output. GrowSmart has been utilised to help farmers overcome their challenges. GrowSmart is a cutting-edge farming technique that uses research on soil qualities, soil types, and crop production statistics to suggest to farmers the best crop to grow as well as which fertiliser to use. By doing this, the likelihood of picking the wrong crop is reduced, and productivity is raised. The purpose of this paper is to propose an IoT and ML based agriculture system that can help farmers or agriculturists in crop prediction based on metrological agriculture theory by getting live metrological data from the crop field using IoT technology and machine learning for prediction, which will enable smart farming and increase their overall yield and quality of products.

Keywords*— *Agriculture, Internet of Things, Machine Learning, Crop Prediction, Recommendation System.

I. INTRODUCTION

A subfield of applied meteorology called "agricultural meteorology" (AM) studies how weather and climate affect agricultural production while also examining the physical surroundings of organisms that are growing plants. A science concerned with the application of meteorology to the measurement and analysis of the physical environment in agricultural systems. The word 'Agro Meteorology' is the abbreviated form of agricultural meteorology which is a study of interaction between meteorological and hydrological factors on the one hand and agriculture in the widest sense. The development of Intelligent Smart Farming devices based on Internet of things and Artificial Intelligence is day by day turning the face of agriculture production by not only enhancing it but also making it cost-effective and reducing wastage [1].

Prices have increased because of structural changes in agriculture and the food system brought on by soaring income and population levels, migration, urbanisation, and speculation. Agriculture is undoubtedly a sector in which the world needs to invest. The goal of advancing agriculture is essential given the global trend towards new technology and their deployment. According to the World Bank, if the rate of population growth in the world remains the same through 2050, we will need to produce 50% more food [2]. However, crop yields may decrease by more than 25% because of climate change. To increase the productivity of the land, attention must be paid to the integration of smart technology into agricultural methods. These studies are carried out under various meteorological and physical conditions, in sites where crops are growing, and in the open air. By collecting linked time series data from sensor networks and lab observations obtained by testing them chemically, Internet of Things (IoT) and AI technologies can reduce the cost and enhance the size of such investigations. The agriculture system that is suggested in this study combines the ideas of machine learning with IOT using IoT boards and a variety of sensors,

CERTICICATES:

Novyi Mir Research Journal

An UGC Care Group-II ISSN No : 0130-7673

Website:https://novyimir.net/

5.3 IMPACT FACTOR

Certificate of Publication

Paper Id : NMRJ/4472

This is to certify that the paper tittled

GrowSmart: An IoT and ML based Intelligent Recommendation System for Crops and Fertilizers

Author by

R M Sai Mohit

From

MITS,Madanapalle Andhra Pradesh.

Has been published in

Novyi Mir Research Journal, Volume 8, Issue 5, May 2023.

Google scholar

Web of Science Group

UGC APPROVED JOURNAL

Kazuhik Hayakawa
Editor - in -Chief
NMRJ

DOI : 16 10098 NMRJ

crossref member

CROSSREF.ORG THE CITATION LINKING BACKBONE

Novyi Mir Research Journal

An UGC Care Group-II ISSN No : 0130-7673

Website:https://novyimir.net/

5.3 IMPACT FACTOR

Certificate of Publication

Paper Id : NMRJ/4472

This is to certify that the paper tittled

GrowSmart: An IoT and ML based Intelligent Recommendation System for Crops and Fertilizers

Author by

M Madhusudhan

From

MITS,Madanapalle Andhra Pradesh.

Has been published in

Novyi Mir Research Journal, Volume 8, Issue 5, May 2023.

Google scholar

Web of Science Group

UGC APPROVED JOURNAL

Kazuhik Hayakawa
Editor - in -Chief
NMRJ

DOI : 16 10098 NMRJ

crossref member

CROSSREF.ORG THE CITATION LINKING BACKBONE

Novyi Mir Research Journal

An UGC Care Group-II ISSN No : 0130-7673

Website:https://novyimir.net/

5.3 IMPACT FACTOR

Certificate of Publication

Paper Id : NMRJ/4472

This is to certify that the paper tittled

GrowSmart: An IoT and ML based Intelligent Recommendation System for Crops and Fertilizers

Author by

B.Jagadeesh

From

MITS,Madanapalle Andhra Pradesh.

Has been published in

Novyi Mir Research Journal, Volume 8, Issue 5, May 2023.

Google scholar

Web of Science Group

UGC APPROVED JOURNAL

Kazuhik Hayakawa
Editor - in -Chief
NMRJ

DOI : 16 10098 NMRJ
crossref member
CROSSREF.ORG THE CITATION LINKING BACKBONE

Novyi Mir Research Journal

An UGC Care Group-II ISSN No : 0130-7673

Website:https://novyimir.net/

5.3 IMPACT FACTOR

Certificate of Publication

Paper Id : NMRJ/4472

This is to certify that the paper tittled

GrowSmart: An IoT and ML based Intelligent Recommendation System for Crops and Fertilizers

Author by

K V Hariprasad

From

MITS,Madanapalle Andhra Pradesh.

Has been published in

Novyi Mir Research Journal, Volume 8, Issue 5, May 2023.

Google scholar

Web of Science Group

UGC APPROVED JOURNAL

Kazuhik Hayakawa
Editor - in -Chief
NMRJ

DOI : 16 10098 NMRJ
crossref member
CROSSREF.ORG THE CITATION LINKING BACKBONE