A PRELIMENERY REPORT ON

PRECISION AGRICULTURE USING MACHINE LEARNING & IOT

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

VENKATA NARAYANA BOMMANABOINA	**71916256E**
ATHARVA LABHASETWAR	**71916316B**
KUNDAN PATIL	**72005231K**

DEPARTMENT OF COMPUTER ENGINEERING

G.H. RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY

WAGHOLI, PUNE 412207

SAVITRIBAI PHULE PUNE UNIVERSITY
2021 -2022

CERTIFICATE

This is to certify that the project report entitles

PRECISION AGRICULTURE USING MACHINE LEARNING& IOT

Submitted by

VENKATA NARAYANA BOMMANABOINA	71916256E
ATHARVA LABHASETWAR	71916316B
KUNDAN PATIL	72005231K

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Mrs. RACHNA SABLE** and it is approved for the partial fulfillment of the requirement of SavitribaiPhule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Mrs. RACHNA SABLE)
Guide
Department of Computer Engineering

(Mrs. RACHNA SABLE)
Head,
Department of Computer Engineering

(Dr. R. D. Kharadkar)
Director,
G .H. Raisoni Institute of Engineering & Technology, Pune - 412207

Place: Pune

Date:

ACKNOWLEDGEMENT

We have great pleasure in presenting report on “PRECISION AGRICULTURE USING MACHINE LEARNING & IOT”. Completing a task is never a one-man effort. It is often a result of invaluable contribution of a number of individuals in direct or indirect manner. We would like to express deepest appreciation towards Dr. R. D. Kharadkar, Principal GHRIET, Pune and Mrs. RACHNA SABLE, HOD Computer Department, whose invaluable guidance supported us in completing this report.

We are profoundly grateful to Mrs. RACHNA SABLE for her expert guidance and continuous encouragement throughout to see that this project report meets the target since from its commencement to completion.

At last, we want to express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

NAME OF THE STUDENTS

VENKATA NARAYANA BOMMANABOINA
ATHARVA LABHASETWAR
KUNDAN PATIL

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. Precision agriculture has been used to solve the farmers' difficulty. Precision agriculture 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 based on site-specific features. This decreases the number of times a crop is chosen incorrectly and increases productivity.

In this paper, this problem is solved by proposing a recommendation system through ML models with majority voting technique using Random Forest, Naive Bayes, Support Vector Machine (SVM), Logistic Regression and Random Forest, as learners to recommend a crop for the site specific parameters with high accuracy and efficiency. In Addition to that we are performing real time testing using IOT system .The fertilizer recommendation system is purely python logic based. In this we compare the data (optimum nutrients for growing the crop) with the user's entered data. Then nutrient having maximum difference is made as HIGH or LOW and according to that suggestions will be fetched.

Keywords: Precision agriculture, Recommendation system, Random Forest, Support Vector Machine (SVM), Logistic Regression.

Table of Contents

List of Figures

CHAPTER 1

1. INTRODUCTION

A farmer's decision about which crop to grow is generally clouded by his intuition and other irrelevant factors like making instant profits, lack of awareness about market demand, overestimating a soil's potential to support a particular crop, and so on. A very misguided decision on the part of the farmer could place a significant strain on his family's financial condition. Perhaps this could be one of the many reasons contributing to the countless suicide cases of farmers that we hear from media on a daily basis. In a country like India, where agriculture and related sector contributes to approximately 20.4 per cent of its Gross Value Added (GVA) , such an erroneous judgment would have negative implications on not just the farmer's family, but the entire economy of a region. For this reason, we have identified a farmer's dilemma about which crop to grow during a particular season, as a very grave one. The need of the hour is to design a system that could provide predictive insights to the Indian farmers, thereby helping them make an informed decision about which crop to grow. With this in mind, we propose a system, an intelligent system that would consider environmental parameters (temperature, rainfall, geographical location in terms of state) and soil characteristics (N, P, K, pH value, soil type and nutrients concentration) before recommending the most suitable crop to the user. In addition to that a fertilizer suggestion is also made which is based on the optimum nutrients of the crops grown.

1.1 Motivation

Agriculture is that the backbone for developing countries like India as quite 70% of population depends on agriculture. Agriculture in India plays a predominant role in economy and employment. The common problem existing among the Indian farmers are they don't choose the proper crop supported their soil requirements and also which fertilizer to be used for his or her crop. thanks to this they face a heavy setback in productivity. This problem of the farmers has been addressed through precision agriculture

1.2 Existing System

More and more researchers have begun to spot this problem in Indian agriculture and are increasingly dedicating their time and efforts to assist alleviate the difficulty. Different works include the employment of Regularized Greedy Forest to see an appropriate crop sequence at a given time stamp. Another approach proposes a model that creates use of historical records of meteorological data as training set. Model is trained to spot climate that are deterrent for the assembly of apples. It then efficiently predicts the yield of apples on the idea of monthly weather patterns.

The use of several algorithms like Artificial Neural Network, K Nearest Neighbors, and Regularized Greedy Forest is demonstrated in [5] to pick out a crop supported the pre-diction yield rate, which, in turn, is influenced by multiple parameters. Additional features included within the system are pesticide prediction and online trading supported agricultural commodities.

1.3 Drawbacks

One shortcoming that we identified all told these notable published works was that the author's of every paper focused on one parameter (either weather or soil) for predicting the suitability of crop growth. However, in our opinion, both these factors should be taken together into consideration concomitantly for the most effective and most accurate prediction. this is often because, a specific soil type could also be it for supporting one variety of crop, but if the climatic conditions of the region don't seem to be suitable for that crop type, then the yield will suffer.

1.4 Proposed System

We to eliminate the aforementioned drawbacks, we propose an efficient Crop Recommendation system- which takes into consideration all the appropriate parameters including temperature, rainfall, location and soil condition, to predict crop suitability. This system is fundamentally concerned with performing the primary function of Agro Consultant, which is providing crop recommendations to farmers. We also provide the fertilizers to be used for crops grown in different states which gives the user an easy and reliable insight to decide and plan the crops

1.5 Plan of Implementation

The steps involved in this system implementation are :-

a) Acquisition of Training Dataset: The accuracy of any machine learning algorithm depends on the number of parameters and the correctness of the training dataset. For the system, we are using various datasets all downloaded for government website and kaggle.

Datasets include:-

Yield dataset, Fertilizer dataset ,Soil nutrient content dataset, Rainfall, Temperature dataset

b). Data Preprocessing: This step includes replacing the null and 0 values for yield by -1 so that it does not effect the overall prediction. Further we had to encode the dataset so that it could be fed into the our ML models.

c). Training ML model : After the preprocessing step we used the dataset to train different machine learning models like Random forest, Decision Tree, Support Vector Machine(SVM) and Logistic regression to attain accuracy as high as possible.

d).Model Evaluation and Saving Model: All the ML models which are trained would be evaluated by comparing their performance (Evaluations Metrics) and Final efficient model is saved using pickle library.

e).Model Exportation and Integration with Webapp: The saved efficient ML model would be integrated with Flask Web Application which would further meant for prediction in user friendly web interface.

f). Real-time Testing of Application: This step includes real-time testing of our whole application using an IOT system which consists of a).Soil NPK Sensor, b).Capacitive Soil Moisture Sensor, c).Temperature Sensor, d).Wireless Transceiver module and e).Arduino Nano board.

Soil NPK sensor, Soil Moisture and Temperature sensors are dipped into soil along with help of Arduino Nano board to acquire all the features of soil.

We get real-time data of soil like N, P, K, Moisture, Temperature, etc which are used to test our pre-built Web Application manually and get the predictions done.

1.6 Problem Definition

In India, agriculture is one of the most important professions. Many of the people do agriculture but are unable to determine which types of crops are more suitable to their soil. Means there are variety of crops which are only suitable for wet soil, some requires medium humidity in the soil to grow but this knowledge is less known to farmers as well as newbies who develop some interest in farming. As of now there are very less resources as well as software's which will help them to improve quality. Such type of software is Precision agriculture using machine learning and IoT.

1.7 Objective of the Project

[1] To build a robust model to provide correct and accurate prediction of crop sustainability in a given state for the particular soil type and climatic conditions.

[2] Provide recommendation of the most effective suitable crops within the area in order that the farmer doesn't incur any losses

[3] Provide fertilizer suggestion for crops supported chemical features.

CHAPTER 2

2. LITERATURE SURVEY

Low-cost IOT + ML design for smart farming with multiple application paper authors Fahad Kamraan Syed, Agniswar Paul, Ajay Kumar, Jaideep Cherukuri in paper [1] proposed system for water management systems and improve current irrigation methods. An IoT and ML-based farming system always keeps farmers aware of the upcoming weather possibilities and gives them the best suggestions about irrigation methods and crops thereby helping in better yield.

In paper [2] author's proposed a smart system that can assist farmers in crop management by considering sensed parameters (temperature, humidity) and other parameters (soil type, location of farm, rainfall) that predicts the most suitable crop to grow in that environment.

Reference Paper [3] determines real time sampling of soil properties using MODIFIED SUPPORT VECTOR REGRESSION, a popular machine learning algorithm and four modules. The Modules include Sensor interfaced to IoT device, Agri cloud, Analyzing the real time sensor data and Agri user interface (AUI). The first module is portable IoT device (NodeMCU) with soil moisture sensor and pH sensor, environmental sensors. Agri cloud module consists of storage. Analyzing the real time data module is processing of types of crops and small plants suggested using modified support vector machine algorithm. Agri-user interface is a basic web interface. Thus, with the help of soil properties farmer will be able to get types of crops and small plants is grown in farmland with help of Modified support vector machine algorithm.

In paper [4] author's proposed new technologies include the use of Internet of Things (IOT) and Machine Learning. The real time data from the field area can be collected using IOT system. The collected data from the field area is fed to the trained model. The trained model then makes the predictions using the data. The result produced by the model greatly helps is sowing the suitable crops in the particular field area.

In Reference paper [5] determines a model is proposed for predicting the soil type and suggest a suitable crop that can be cultivated in that soil. The model has been tested using various machine learning algorithms such as KNN, SVM and logistic regression. The accuracy of the present model is maximum than the existing models.

Aruul Mozhi Varman S proposed an IOT and deep learning based smart agriculture systems. This system monitors and collects the soil parameters from the field with the help of a wireless sensor network. The collected data is then uploaded in the cloud. Finally, the systems suggest best irrigation practices to the farmers by predicting the crop to be sown for next crop rotation. This information will be sent as an SMS to the farmers. The parameters include soil temperature, atmospheric temperature, and humidity [6]. This system suggests further improving the effectiveness by predicting the suitable time for applying pesticides, fertilizer, and manures.

In paper [7] proposed a system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. The ML and IoT based suggestions will significantly educate the farmer and help them minimize costs and make strategic decisions by replacing intuition and passed-down knowledge with far more reliable data-driven ML models. This allows for a scalable, reliable solution to an important problem affecting hundreds of millions of people

CHAPTER 3

Theoretical Background

3.1 Overview on Machine Learning

Machine learning is an application of artificial intelligence (AI) that gives systems the ability to automatically learn and evolve from experience without being specially programmed by the programmer. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The main aim of machine learning is to allow computers to learn automatically and adjust their actions to improve the accuracy and usefulness of the program, without any human intervention or assistance. Traditional writing of programs for a computer can be defined as automating the procedures to be performed on input data in order to create output artifacts. Almost always, they are linear, procedural and logical. A traditional program is written in a programming language to some specification, and
it has properties like:

- We know or can control the inputs to the program.
- We can specify how the program will achieve its goal.
- We can map out what decisions the program will make and under what conditions it makes them.
- Since we know the inputs as well as the expected outputs, we can be confident that the program will achieve its goal.

Traditional programming works on the premise that, as long as we can define what a **program needs to do, we are confident we can define how a program can achieve that** goal. This is not always the case as sometimes, however, there are problems that you can represent in a computer that you cannot write a traditional program to solve.
Such problems resist a procedural and logical solution. They have properties such as:

- The scope of all possible inputs is not known beforehand.
- You cannot specify how to achieve the goal of the program, only what that goal is.
- You cannot map out all the decisions the program will need to make to achieve its goal.

- You can collect only sample input data but not all possible input data for the program.

3.1.1 Supervised and Unsupervised Learning

Machine learning techniques can be broadly categorized into the following types: Supervised learning takes a set of feature/label pairs, called the training set. From this training set the system creates a generalized model of the relationship between the set of descriptive features and the target features in the form of a program that contains a set of rules. The objective is to use the output program produced to predict the label for a previously unseen, unlabelled input set of features, i.e. to predict the outcome for some new data. Data with known labels, which have not been included in the training set, are classified by the generated model and the results are compared to the known labels. This dataset is called the test set. The accuracy of the predictive model can then be calculated as the proportion of the correct predictions the model labeled out of the total number of instances in the test set.

Unsupervised learning takes a dataset of descriptive features without labels as a training set. In unsupervised learning, the algorithms are left to themselves to discover interesting structures in the data. The goal now is to create a model that finds some hidden structure in the dataset, such as natural clusters or associations. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system does not figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data. Unsupervised learning can be used for clustering, which is used to discover any inherent grouping that are already present in the data. It can also be used for association problems, by creating rules based on the data and finding relationships or associations between them.

Semi-supervised machine learning falls somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring labeled data generally does not require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Machine learning algorithms are tools to automatically make decisions from data in order to achieve some over-arching goal or requirement. The promise of machine learning is that it can solve complex

problems automatically, faster and more accurately than a manually specified solution, and at a larger scale. Over the past few decades, many machine learning algorithms have been developed by researchers, and new ones continue to emerge and old ones modified.

3.2 Machine Learning Tools

There are many different software tools available to build machine learning models and to apply these models to new, unseen data. There are also a large number of well defined machine learning algorithms available. These tools typically contain libraries implementing some of the most popular machine learning algorithms. They can be categorized as follows :

- Pre-built application-based solutions.
- Programming languages which have specialized libraries for machine learning.

Using programming languages to develop and implement models is more flexible and gave us better control of the parameters to the algorithms. It also allows us to have a better understanding of the output models produced. Some of the popular programming languages used in the field of machine learning are:

- **Python**: Python is an extremely popular choice in the field of machine learning and AI development. Its short and simple syntax make it extremely easy to learn
- **R**: R is one of the most effective and efficient languages for analyzing and manipulating data in statistics. Using R, we can easily produce well-designed publication-quality plot, including mathematical symbols and formulae where needed. Apart from being a general purpose language, R has numerous of packages like RODBC, G models, Class and Tm which are used in the field of machine learning. These packages make the implementation of machine learning algorithms easy, for cracking the business associated problems
- **Tensorflow:** TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide

variety of other domains, as well. TensorFlow provides stable Python and C++ APIs, as well as non-guaranteed backward compatible API for other languages.

3.3 SciKit-learn

SciKit learn is an open source machine learning library built for python. Since its release in 2007, Scikit-learn has become one of the most popular open source machine learning libraries. Scikit-learn (also called sklearn) provides algorithms for many machine learning tasks including classification, regression, dimensionality reduction and clustering. The documentation for scikit-learn is comprehensive, popular and well maintained. Sklearn is built on mature Python Libraries such as NumPy, SciPy, and matplotlib. While languages such as R and MATLAB are extremely popular and useful for machine learning, we decided to choose Python along with its SciKit-learn libraries as our programming language of choice. The reasons for this are:

- We already have some familiarity and exposure to Python, and thus have a smaller learning curve.
- Both Python and Scikit-learn have excellent documentation and tutorials available online
- The number of classic machine learning algorithms that come with Scikit-learn, and the consistent patterns for using the different models i.e., each model can be used with the same basic commands for setting up the data, training the model and using the model for prediction. This makes it easier to try a range of machine learning algorithms on the same data.
- The machine learning algorithms included with sklearn have modifiable parameters known as hyper parameters that effect the performance of the model. These usually have sensible default values, so that we can run them without needing a detailed knowledge or understanding of their semantics.
- The IPython notebook, which is an interactive computational environment for Python, in which a user can combine code execution, rich text, mathematics and plots in a web page. This functionality allows us to provide the notebooks we used to run our experiments almost as an audit and in a presentable.

3.4 Dataset

For the system, we are using various datasets all downloaded for government website and kaggle.

Datasets include:-

- Yield dataset
- Fertilizer dataset
- Soil nutrient content dataset
- Rainfall Temperature dataset

A brief description of the datasets:

- **Yield Dataset:** This dataset contains yield for 16 major crops grown across all the states in kg per hectare. Yield of 0 indicates that the crop is not cultivated in the respective state.
- **Soil nutrient content dataset :** This dataset has columns with the attributes in the order-State, Nitrogen content, Phosphorous content, Potassium content and average ph
- **Rainfall Temperature dataset**: This dataset contains crops, max and min rainfall, max and min temperature, max and min rainfall and ph values.

3.5 Data Preprocessing

This step includes replacing the null and 0 values for yield by -1 so that it does not effect the overall prediction. Further we had to encode the data-set so that it could be fed into the ML models.

Chapter 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 Introduction

This Software Requirements Specification provides a complete description of all the Functions and constraints of the "Precision agriculture using machine learning & IOT". The document, describes the issue related to the agriculture system and what actions are needed to perform to grow which type of crop should be grown in your soil and your environment to get better crop production.

The basic idea of this project comes from the farmers who don't know which type of crop will be better for their soil according to nutrition and other environment conditions. In India, agriculture is one of the most important professions. Many of the people do agriculture but are unable to determine which types of crops are more suitable to their soil. Means there are variety of crops which are only suitable for wet soil, some requires medium humidity in the soil to grow but this knowledge is less known to farmers as well as newbies who develop some interest in farming. As of now there are very less resources as well as software's which will help them to improve quality. Such types of software are precision agriculture using machine learning.

4.1.1 Project Scope

- Improve farm management efficiency by adjusting field/crop treatments
- Getting a better result for which type of crop will be growing on your agriculture field.
- Getting more productivity from less efforts by using our application
- Improve farm management efficiency by adjusting field/crop treatments
- Optimise efforts and resources, reduce consumption and waste, and boost land productivity.
- Which type of fertilizers should be used if any crop having any disease we can minimize using our app

This are some project scope of our project.

4.1.2 User Classes and Characteristics

This protocol is implemented in python language .We also use flask, bootstrap frameworks. The system design has done using python flask web application.

This project is for end user who has to know about which type of crop should be better yield in their agriculture field just like farmer and newbies who don't know how to do agriculture in their filed so our project provide all necessary thing to get better crop productivity in your field. Also we can minimize the effect of disease of your crop by identifying which type of disease it is.

4.1.3 Assumptions and Dependencies

- Data is an asset. It is valuable resource, as it has real and measurable value. Accurate and timely data is critical to quality and efficiency of service. Data input and its accuracy will depend on the user. Accountability of data will be defined
- Content generation and updation will be done timely by the user
- Portal Management Framework will be devised and user will play active role in it
- User will provide content in local languages

4.2 Functional Requirements

- System must be fast and efficient
- User friendly GUI
- Performance
- System Validation input
- Proper output

4.2.1 System Feature 1(Functional Requirement)

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

1. All the data must be in the same format as a structured data.
2. The data collected will be vectorized and sent across to the classifier.

4.3 External Interface Requirements

4.3.1 User Interfaces

- Front End Software: Flask Framework integrated with HTML, CSS, JS, BOOTSTRAP
- Back End Software: Machine Learning (Python)

4.3.2 Hardware Interfaces

- RAM - Minimum 512 MB.
- Processor - i3 or above and above with 2.5 GHz

4.3.3 Software Interfaces

- OS: Ubuntu, Windows, Mac
- Tools: VScode or Python IDE and Jupyter Notebook.
- Programming Language: Python flask, HTML, CSS, JS, BOOTSTRAP.
- Dataset: A Dataset which is openly available in kaggle.
- Libraries/Tools : Seaborn, Pandas, NumPy, SciKit-Learn, Pytorch, ResNet-9, SQLAlchemy, Pickle

4.3.4 Communication Interfaces

The communication protocol started from http from UI interface in which person can see some details of the soil nutrition section by filling up details user can see which type of crop should be grown in your region and your agriculture field. And another one is person can just upload the picture of crop then user will get information about the diseases of crop.

4.4 Nonfunctional Requirements

Nonfunctional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Nonfunctional

requirements arise through the user needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

4.4.1 Performance Requirements

The System should be interactive and performance should be efficient. The delays of the system should be less. Also a stable internet connection is required.

4.4.2 Safety Requirements

This Specification shall be sufficient detailed to allow the design and implement to achieve the required safety integrity and allow an assessment of functional safety

4.4.3 Security Requirements

We provide login system by passwords for each level of access

4.4.4 Software Quality Attributes

Availability:

The information regarding the nutrition's must be available.

Usability:

System should be interactive and easy to understand.

Maintainability:

System should be easy to maintain.

Robustness:

Single application failure will not affect the system. System should be fault tolerant. System should reliable enough to sustain in any condition. It should give Perfect results.

4.5 System Requirements

4.5.1 Database Requirements

MYSQL

4.5.2 Software Requirements

- Operating System: Windows, Linux and Mac
- Coding Language: Python Flask, HTML,CSS,JS,BOOSTRAP
- Tools: Seaborn, Pandas, NumPy, SciKit-Learn, Pytorch, ResNet-9, SQLAlchemy, Pickle

4.5.3 Hardware Requirements

- Processor: i3 and above with 2.5 GHz.
- RAM: 512 MB.
- Hard disk space: = 8GB

Other Requirements:

The below hardware instruments are used for testing our model in real-time using IOT system which has following set of instruments.

- Soil NPK Sensor
- DS18B20 Temperature Sensor
- Capacitive Soil Moisture Sensor
- NRF24L01 Wireless Transceiver
- Arduino Nano Board

CHAPTER 5

System Analysis

5.1 Feasibility Study

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. Since the project is Machine learning based, the cost spent in exectuating this project would not demand cost for softwares and related products, as most of the products are open source and free to use. Hence the project would consumed minimal cost and is economically feasible.

5.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Since machine learning algorithms is based on pure math there is very less requirement for any professional software. And also most of the tools are open source. The best part is that we can run this software in any system without any software requirements which makes them highly portable. Also most of the documentation and tutorials make easy to learn the technology

5.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The main purpose of this project which is based on crop prediction is to prevent the farmer from incurring losses and improve productivity. This also ensures that there is no scarcity of food as lack of production may lead to severe consequences. Thus, this is a noble cause for the sake of the society, a small step taken to achieve a secure future.

5.2 Analysis

5.2.1 Performance Analysis

Most of the software we use is open source and free. The models whichwe use in this software, learn only once ,i.e once they are trained theyneed not be again fed in for the training phase. One can directly predictfor values, hence time-complexity is very less. Therefore this model istemporally sound.

5.2.2 Technical Analysis

As mentioned earlier, the tools used in building this software is opensource. Each tool contains simple methods and the required methodsare overridden to tackle the problem.

5.2.3 Economical Analysis

The completion of this project can be considered free of cost in itsentirety. As the software used in building the model is free of cost andall the data sets used are being downloaded from kaggle and Govt. ofIndia website.

CHAPTER 6

System Design

6.1 Analysis Models: SDLC Model to be applied

The waterfall model is a sequential software development process, in which progress is seen as owing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis:

This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

System Design:

Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:- algorithm, data structure, software architecture etc.

Coding:

In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine.

Implementation:

The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executable, user manuals and additional software documentation.

Testing:

In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation

Maintenance:

The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer needs, adapt to accommodate changes in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software

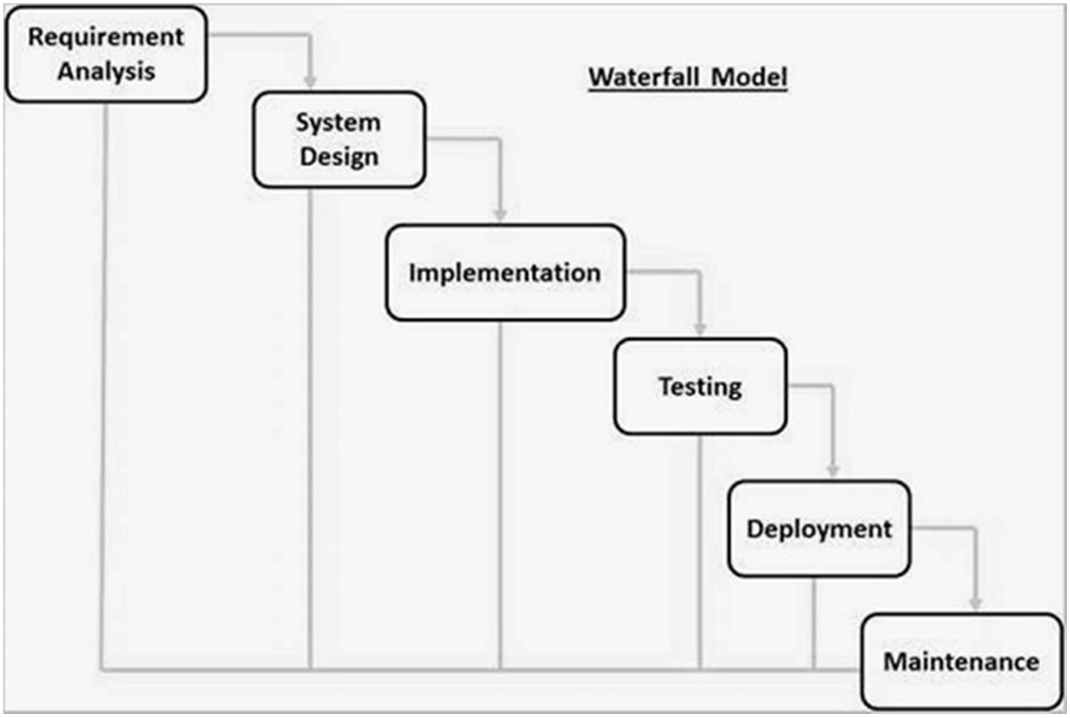


Fig-1 Waterfall Model

6.2. Advantages of Waterfall model

- _ Clear project objective
- _ Stable project requirements
- _ Progress of system is measurable.
- _ Logic of software development is clearly understood.
- _ Better resource allocation.

6.3 System Architecture

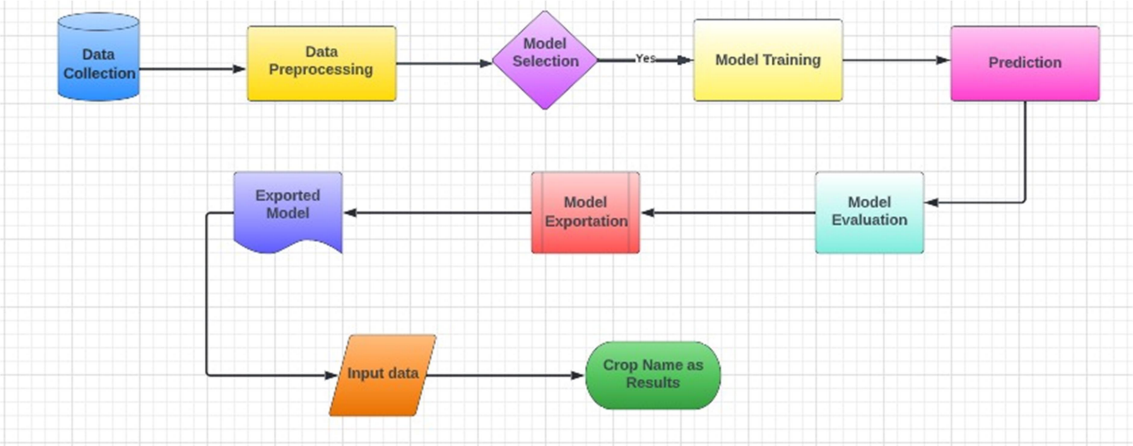


Fig-2 System Architecture

A system architecture is a conceptual model using which we can define the structure and behaviour of that system. It is a formal representation of a system. Depending on the context, system architecture can be used to refer to either a model to describe the system or a method used to build the system. Building a proper system architecture helps in analysis of the project, especially in the early stages. Figure 6.3 depicts the system architecture and is explained in the following section.

6.4 Sequence Diagram

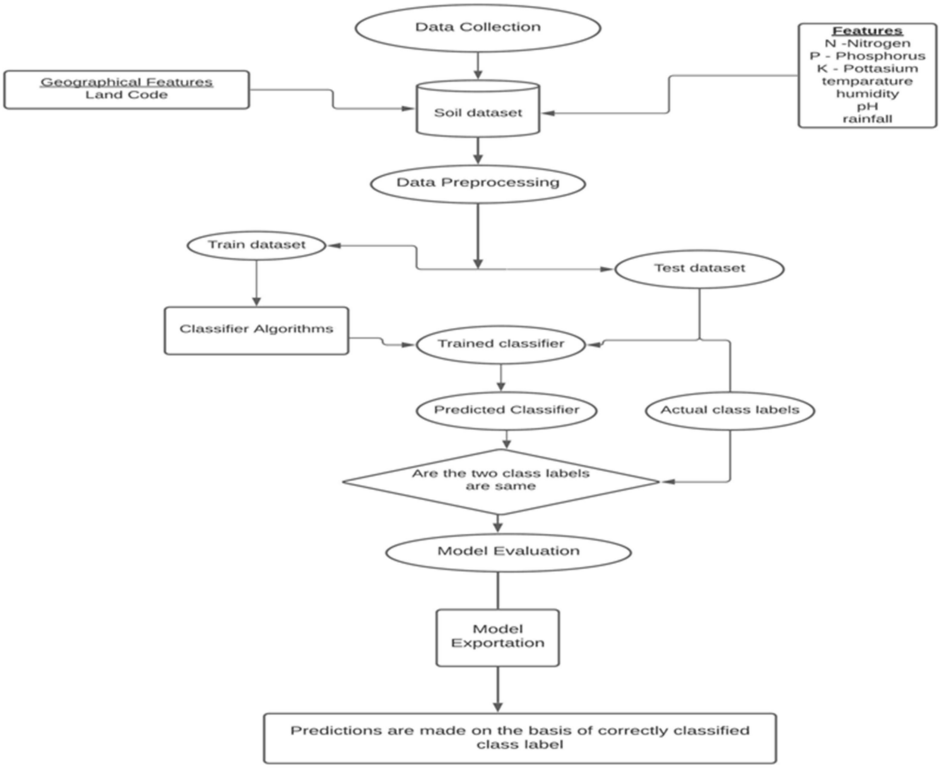


Fig-3 Sequence Diagram

6.5 Data Flow Diagrams

6.5.1 DFD Level-0

Fig-4 DFD Level-0

6.5.2 DFD Level-1

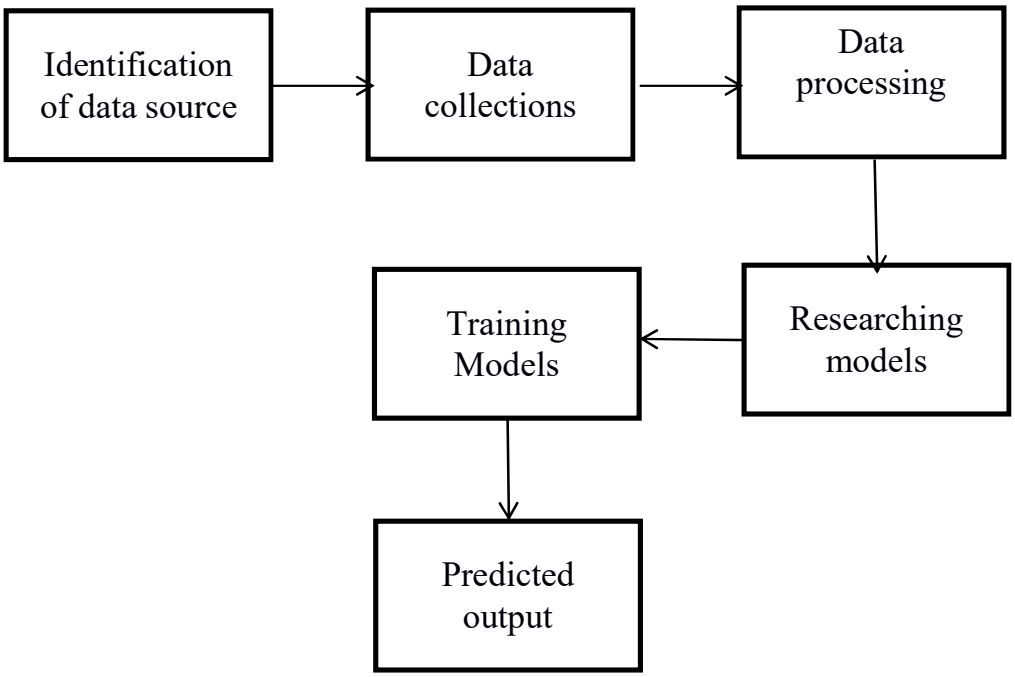


Fig-5 DFD Level-1

6.6 UML Diagrams

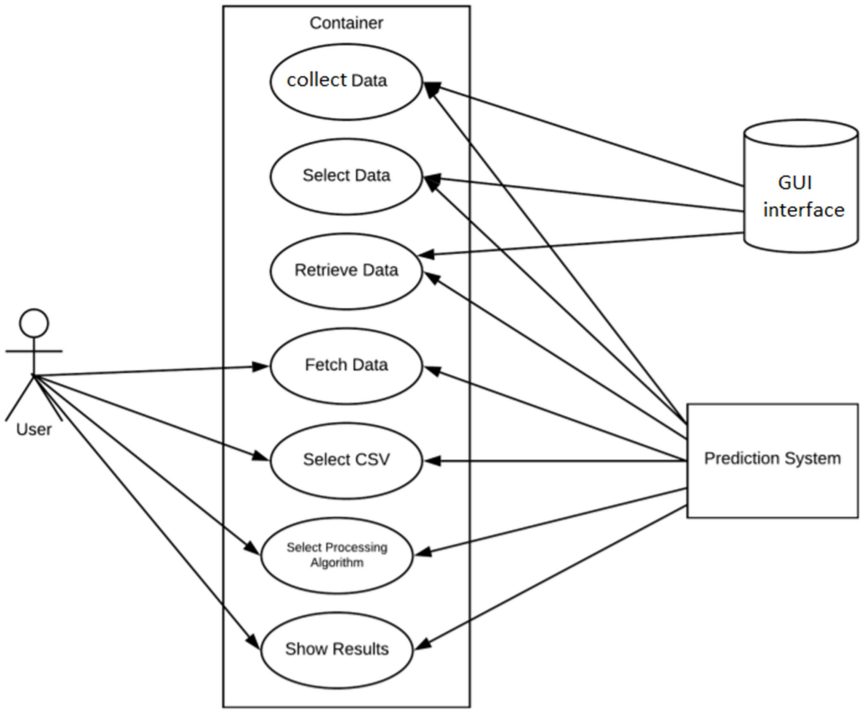


Fig-6 UML diagram

CHAPTER 7

Implementation

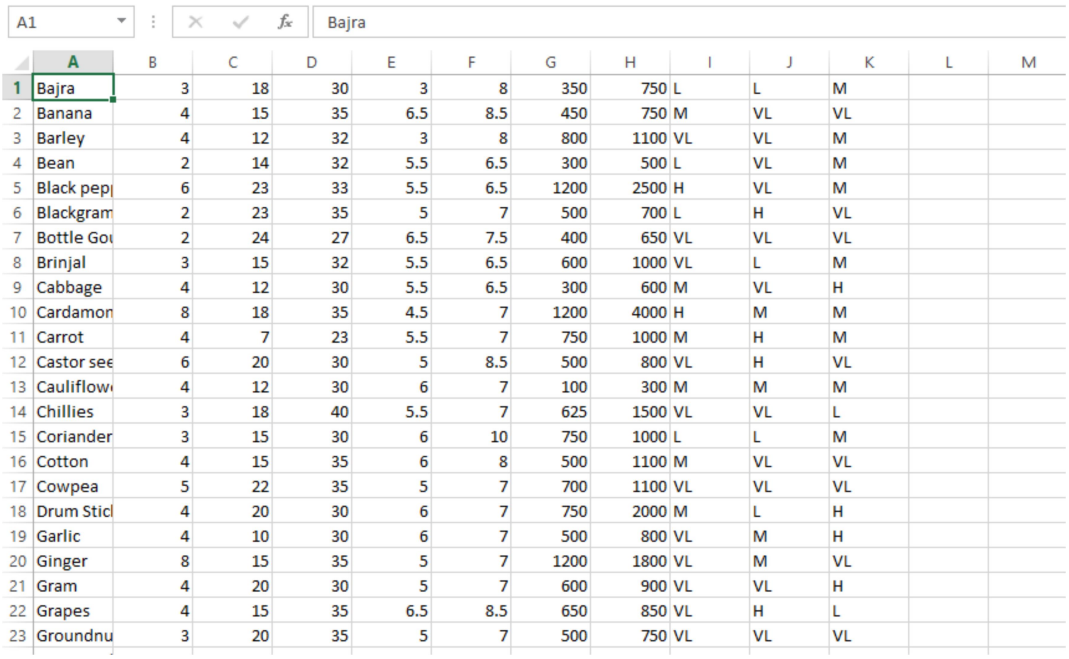
7.1 Data Analysis

One of the first steps we perform during implementation is an analysis of the data. This was done by us in an attempt to find the presence of any relationships between the various attributes present in the dataset. Acquisition of Training Dataset: The accuracy of any machine learning algorithm depends on the number of parameters and the correctness of the training dataset. We In this project analyzed multiple datasets collected from Government website -https://data.gov.in/ and Kaggle and carefully selected the parameters that would give the best results. Many work done in this field have considered environmental parameters to predict crop sustainability some have used yield as major factor where as in some works only economic factors are taken into consideration. We have tried to combine both environmental parameters like rainfall, temperature, ph, nutrients in soil, soil type, location and economic parameters like production, and yield to provide accurate and reliable recommendation to the farmer on which crop will be most suitable for his land.

A1 | state

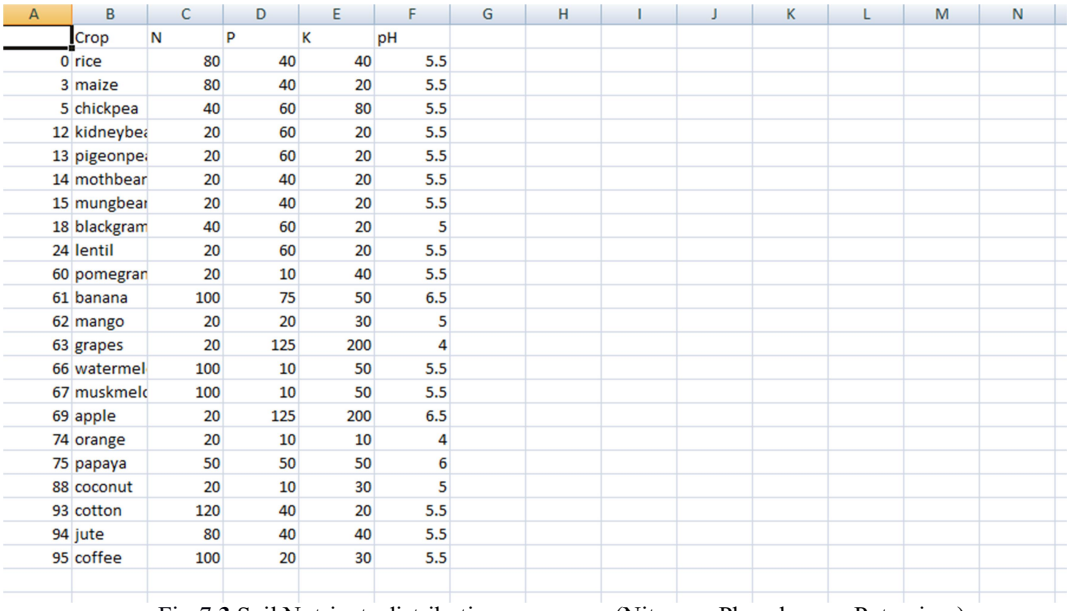
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	state	Rice	Jowar	Bajra	Maize	Ragi	Wheat	Barley	Gram	Tur	Groundnu	Mustard	Soyabean	Sunflowe	Cotton	Jute	Mesta	Sugarcan
2	Andhra Pr	2921.186	1054.115	906.734	3328.871	1202.798	792.5525	0	1139.872	441.4852	852.6383	356.25	1450.464	768.493	337.1242	0	1507.329	76472.3
3	Arunachal	1190.479	0	0	1365.529	0	1641.025	0	0	1000	0	963.015	1258.428	0	0	0	0	19073.6
4	Assam	1449	0	0	710	0	1150	0	519	707	0	508	0	0	106	1736	909	3794
5	Bihar	1300.151	929.1806	933.166	2281.541	794.861	1876.421	1124.002	938.5558	1183.194	652.7778	853.5547	0	1365.018	0	1474.782	1344.15	43134.9
6	Chattisgar	1177	843	0	1567	263	996	888	728	471	1122	369	857	459	136	0	361	255
7	Goa	2652	0	0	0	1000	0	0	0	0	1813	0	0	0	0	0	0	5216
8	Gujarat	1610	901	1152	1300	868	2451	0	739	796	1161	1341	716	0	373	0	0	7405
9	Haryana	2894	296	1313	2228	0	3979	2735	725	988	809	1304	0	1598	452	0	0	5998
10	Himachal	1447	0	0	2251	1104	1482	1207	901	0	0	495	1342	0	0	0	0	1801
11	Jammu &	1960	589	571	1535	0	1543	631	0	0	0	635	0	0	0	0	0	
12	Jharkhanc	1413	988	1253	1465	632	1682	922	886	860	698	558	0	0	0	0	1051	3460
13	Karnataka	2561.393	845.2645	626.1794	2654.594	1492.443	736.6376	0	506.7264	489.3979	696.5809	278.5281	683.0272	456.3873	219.2412	0	265.7269	83235.9
14	Kerala	2197	490	0	0	1070	0	0	0	0	763	0	0	0	250	0	0	9136
15	Madhya P	862	985	1244	1525	351	1630	1228	855	754	992	925	928	453	164	0	382	3893
16	Maharash	1594	812	695	1835	992	1320	637	614	683	1066	317	1175	534	189	0	273	7520
17	Manipur	2315.246	0	0	2495.178	0	0	0	0	0	0	461.1111	0	0	0	0	0	32206.8
18	Meghalay	1692	0	0	1452	0	1699	0	0	769	0	648	945	0	172	1430	835	
19	Mizoram	1501	0	0	1814	0	0	0	0	0	0	742	1113	0	368	0	0	407
20	Nagaland	1556	1246	1269	1609	0	1716	1756	1027	992	1308	842	1282	1197	570	587	0	4475
21	Orissa	1366.031	608.9923	559.4005	1412.742	641.3716	1445.9	0	627.3451	704.1701	1111.208	205.4544	769.2308	802.4356	327.1654	1797.091	796.46	60066.4
22	Puniab	3686	0	984	2702	0	4259	3309	892	876	868	1105	0	1602	563	0	0	6027

Fig-7.1 Yield Dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Bajra	3	18	30	3	8	350	750	L	L	M		
2	Banana	4	15	35	6.5	8.5	450	750	M	VL	VL		
3	Barley	4	12	32	3	8	800	1100	VL	VL	M		
4	Bean	2	14	32	5.5	6.5	300	500	L	VL	M		
5	Black pep	6	23	33	5.5	6.5	1200	2500	H	VL	M		
6	Blackgram	2	23	35	5	7	500	700	L	H	VL		
7	Bottle Go	2	24	27	6.5	7.5	400	650	VL	VL	VL		
8	Brinjal	3	15	32	5.5	6.5	600	1000	VL	L	M		
9	Cabbage	4	12	30	5.5	6.5	300	600	M	VL	H		
10	Cardamon	8	18	35	4.5	7	1200	4000	H	M	M		
11	Carrot	4	7	23	5.5	7	750	1000	M	H	M		
12	Castor see	6	20	30	5	8.5	500	800	VL	H	VL		
13	Cauliflow	4	12	30	6	7	100	300	M	M	M		
14	Chillies	3	18	40	5.5	7	625	1500	VL	VL	L		
15	Coriander	3	15	30	6	10	750	1000	L	L	M		
16	Cotton	4	15	35	6	8	500	1100	M	VL	VL		
17	Cowpea	5	22	35	5	7	700	1100	VL	VL	VL		
18	Drum Stic	4	20	30	6	7	750	2000	M	L	H		
19	Garlic	4	10	30	6	7	500	800	VL	M	H		
20	Ginger	8	15	35	5	7	1200	1800	VL	M	VL		
21	Gram	4	20	30	5	7	600	900	VL	VL	H		
22	Grapes	4	15	35	6.5	8.5	650	850	VL	H	L		
23	Groundnu	3	20	35	5	7	500	750	VL	VL	VL		

Fig-7.2 Temperature Rainfall and Nutrients dataset



	Crop	N	P	K	pH
0	rice	80	40	40	5.5
3	maize	80	40	20	5.5
5	chickpea	40	60	80	5.5
12	kidneybea	20	60	20	5.5
13	pigeonpea	20	60	20	5.5
14	mothbear	20	40	20	5.5
15	mungbear	20	40	20	5.5
18	blackgram	40	60	20	5
24	lentil	20	60	20	5.5
60	pomegran	20	10	40	5.5
61	banana	100	75	50	6.5
62	mango	20	20	30	5
63	grapes	20	125	200	4
66	watermel	100	10	50	5.5
67	muskmelo	100	10	50	5.5
69	apple	20	125	200	6.5
74	orange	20	10	10	4
75	papaya	50	50	50	6
88	coconut	20	10	30	5
93	cotton	120	40	20	5.5
94	jute	80	40	40	5.5
95	coffee	100	20	30	5.5

Fig-7.**3** Soil Nutrients distribution as per crop(Nitrogen,Phosphorous,Potassium).

7.2 Data Preprocessing

After analyzing and visualizing the data, the next step is preprocessing.

Data preprocessing is an important step as it helps in cleaning the data and making it suitable for use in machine learning algorithms. Most of the focus in preprocessing is to remove any outliers or erroneous data, as well as handling any missing values.

Missing data can be dealt with in two ways. The first method is to simply remove the entire row which contains the missing or error value. While this an easy to execute method, it is better to use only on large datasets. Using this method on small datasets can reduce the dataset size too much, especially if there are a lot of missing values.

This can severely affect the accuracy of the result. Since ours is a relatively small dataset, we will not be using this method. The dataset that we used had values that were in string format so we had to transform and encode the into integer valued so as to pass as an input to the neural network. First we converted the data into pandas categorical data and then generated codes for crops and states respectively we than appended these and created separated datasets.

```
import pandas as pd
# Reading the data
crop_data_path = '../Data-raw/cpdata.csv'
fertilizer_data_path = '../Data-raw/Fertilizer.csv'
crop = pd.read_csv(crop_data_path)
fert = pd.read_csv(fertilizer_data_path)
# Function for lowering the cases
defchange_case(i):
    i = i.replace(" ", "")
    i = i.lower()
    return i
fert['Crop'] = fert['Crop'].apply(change_case)
crop['label'] = crop['label'].apply(change_case)
crop_names = crop['label'].unique()
print(crop_names)
crop_names_from_fert = fert['Crop'].unique()
print(crop_names_from_fert)
for i in crop_names_from_fert:
    print(crop[crop['label'] == i])
extract_labels = []
for i in crop_names_from_fert:
    if i in crop_names:
```

```
extract_labels.append(i)
# using extract labesl on crop to get all the data related to those labels
new_crop = pd.DataFrame(columns = crop.columns)
new_fert = pd.DataFrame(columns = fert.columns)
for label in extract_labels:
new_crop = new_crop.append(crop[crop['label'] == label])
for label in extract_labels:
new_fert = new_fert.append(fert[fert['Crop'] == label].iloc[0])
print(new_crop)
print(new_fert)
new_crop.to_csv('../Data-raw/MergeFileCrop.csv')
new_fert.to_csv('../Data-raw/FertilizerData.csv')
```

7.3 Machine Learning Models

7.3.1 Decision Tree :

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the

```
decision rules and the fitter the model.
features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = df['label']
#features = df[['temperature', 'humidity', 'ph', 'rainfall']]
labels = df['label']
# Initialzing empty lists to append all model's name and corresponding name
acc = []
model = []
# Splitting into train and test data
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random_state =2)
from sklearn.tree import DecisionTreeClassifier
DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)
DecisionTree.fit(Xtrain,Ytrain)
predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)
print(classification_report(Ytest,predicted_values))
from sklearn.model_selection import cross_val_score
# Cross validation score (Decision Tree)
```

```
score = cross_val_score(DecisionTree, features, target,cv=5)
print(score)
#Saving trained Decision Tree model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
DT_pkl_filename = '../models/DecisionTree.pkl'
# Open the file to save as pkl file
DT_Model_pkl = open(DT_pkl_filename, 'wb')
pickle.dump(DecisionTree, DT_Model_pkl)
# Close the pickle instances
DT_Model_pkl.close()
```

7.3.2 Support Vector Machine

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

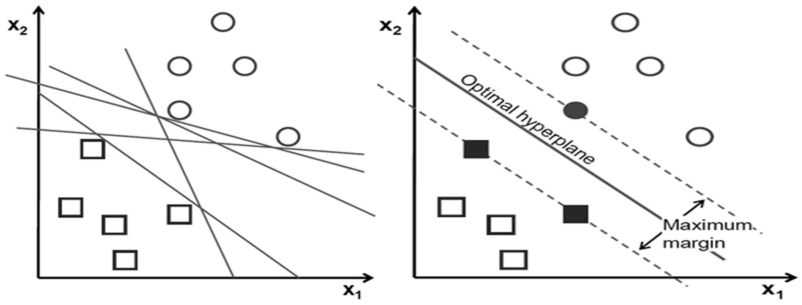


Fig-8 support vector machine

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

```
from sklearn.svm import SVC
SVM = SVC(gamma='auto')
SVM.fit(Xtrain,Ytrain)
predicted_values = SVM.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
score = cross_val_score(SVM,features,target,cv=5)
print(score)
```

7.3.3 Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

What are the types of logistic regression

1. Binary (eg. Tumor Malignant or Benign)
2. Multi-linear functions failsClass (eg. Cats, dogs or Sheep's)

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

```
from sklearn.linear_model import LogisticRegression
LogReg = LogisticRegression(random_state=2)
LogReg.fit(Xtrain,Ytrain)
predicted_values = LogReg.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
score = cross_val_score(LogReg,features,target,cv=5)
print(score)
import pickle
# Dump the trained Naive Bayes classifier with Pickle
LR_pkl_filename = '../models/LogisticRegression.pkl'
# Open the file to save as pkl file
LR_Model_pkl = open(DT_pkl_filename, 'wb')
pickle.dump(LogReg, LR_Model_pkl)
# Close the pickle instances
LR_Model_pkl.close()
```

7.3.4 Random Forest

Random forest is a ***Supervised Machine Learning Algorithm*** that is ***used widely in Classification and Regression problems***. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing ***continuous variables*** as in the case of regression and ***categorical variables*** as in the case of classification. It performs better results for classification problems.

Random Forests was developed specifically to address the problem of high-variance in Decision Trees. Like the name suggests, you're not training a single Decision Tree, you're training an entire forest! In this case, a forest of Bagged Decision Trees.

Random Forests algorithm follows these steps:

1. Take the original dataset and create *N* bagged samples of size *n*, with *n* smaller than the original dataset.

2. Train a Decision Tree with each of the *N* bagged datasets as input. But, when doing a node split, don't explore all features in the dataset. Randomly select a smaller number, *M* features, from all the features in training set. Then pick the best split using impurity measures, like Gini Impurity or Entropy.

3. Aggregate the results of the individual decision trees into a single output.

4. Average the values for each observation, produced by each tree, if you're working on a Regression task.

5. Do a majority vote across all trees, for each observation, if you're working on a Regression task.

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)
predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
# Cross validation score (Random Forest)
score = cross_val_score(RF,features,target,cv=5)
print(score)
import pickle
# Dump the trained Naive Bayes classifier with Pickle
RF_pkl_filename = '../models/RandomForest.pkl'
# Open the file to save as pkl file
RF_Model_pkl = open(RF_pkl_filename, 'wb')
pickle.dump(RF, RF_Model_pkl)
# Close the pickle instances
RF_Model_pkl.close()
```

CHAPTER 8

Testing

8.1 Testing Methodologies

The program comprises of several algorithms which are tested individually for the accuracy. we check for the correctness of the program as a whole and how it performs.

8.2 Unit Testing

Unit tests focus on ensuring that the correct changes to the world state take place when a transaction is processed. The business login transaction processor functions should have unit tests, ideally with 100 percent code coverage. This will ensure that you do not have typos or logic errors in the business logic. The various modules can be individually run from a command line and tested for correctness. The tester can pass various values, to check the answer returned and verify it with the values given to him/her. The other work around is to write a script, and run all the tests using it and write the output to a log _le and using that to verify the results. We tested each of the algorithms individually and made changes in preprocessing accordingly to increase the accuracy.

8.3 System Testing

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing is the testing of a complete and fully integrated software product and White Box Testing. System test falls under the black box testing category of software testing. Different Types of System Testing:

- **Usability Testing** - Usability Testing mainly focuses on the users ease to use the application, exibility in handling controls and ability of the system to meet its objectives.
- **Load Testing** - Load Testing is necessary to know that a software solution will perform under real-life loads.
- **Regression Testing-** - Regression Testing involves testing done to make sure none of the changes made over the course of the development process have caused new bugs.
- **Recovery Testing** - Recovery testing is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.

- **Migration Testing** - Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

8.4 Quality Assurance

Quality Assurance is popularly known as QA Testing, is defined as an activity to ensure that an organization is providing the best possible product or service to customers. QA focuses on improving the processes to deliver Quality Products to the customer. An organization has to ensure, that processes are efficient and effective as per the quality standards defined for software products.

8.5 Functional Test

Functional Testing is also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. As chat-bot evolves into new application areas, functional testing of essential chatbot components. Functional testing evaluates use-case scenarios and related business processes, such as the behavior of smart contracts.

CHAPTER 9
Results
9.1 Algorithms and Their Accuracy

For the purposes of this project we have used four popular algorithms:
Decision Trees, Logistic regression, Support Vector Machine and Random Forest. All the algorithms are based on supervised learning. Our overall system is divided
into two modules:

- Crop recommender
- Fertilizer Recommender/Suggestion

Algorithm	Accuracy (%)
Decision Tree	90%
SVM	97%
Logistic Regression	95%
Random Forest	99%

Accuracy Comparison of ML Models:

```
plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
```

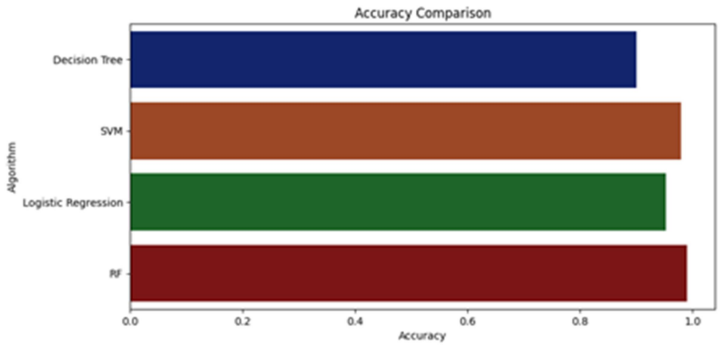
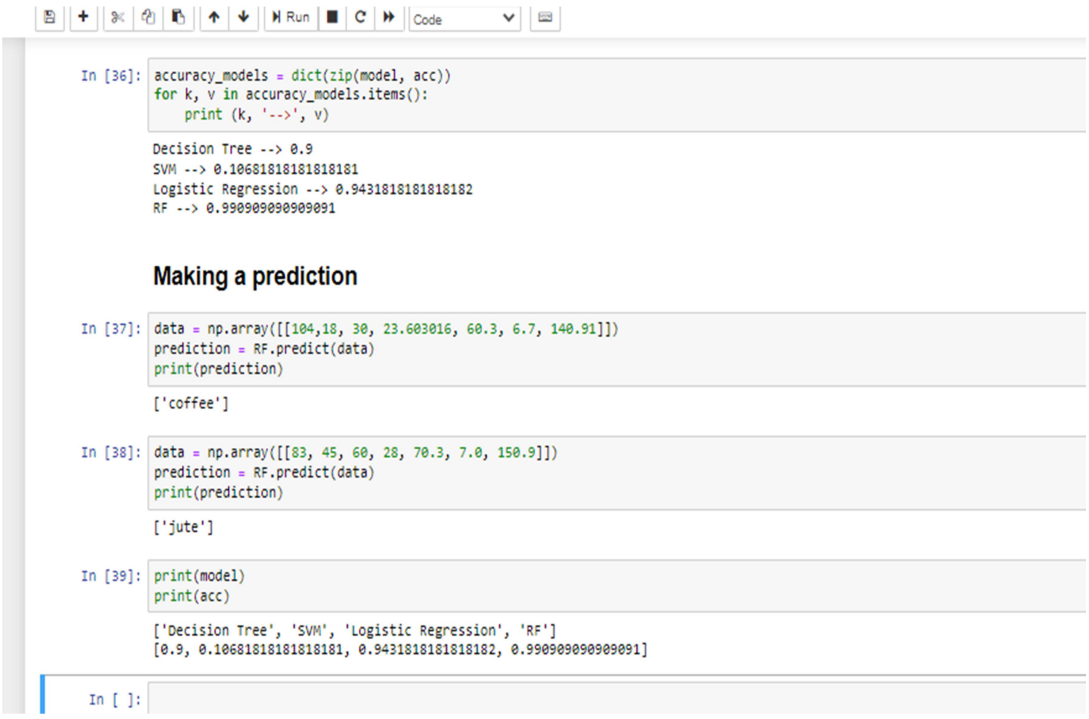


Fig-9 Accuracy Comparison

Hence, **Random Forest** is our Final efficient model.

Output for Crop recommender:



```
In [36]: accuracy_models = dict(zip(model, acc))
         for k, v in accuracy_models.items():
             print (k, '-->', v)

Decision Tree --> 0.9
SVM --> 0.10681818181818181
Logistic Regression --> 0.9431818181818182
RF --> 0.990909090909091
```

Making a prediction

```
In [37]: data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
         prediction = RF.predict(data)
         print(prediction)

['coffee']
```

```
In [38]: data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
         prediction = RF.predict(data)
         print(prediction)

['jute']
```

```
In [39]: print(model)
         print(acc)

['Decision Tree', 'SVM', 'Logistic Regression', 'RF']
[0.9, 0.10681818181818181, 0.9431818181818182, 0.990909090909091]
```

```
In [ ]:
```

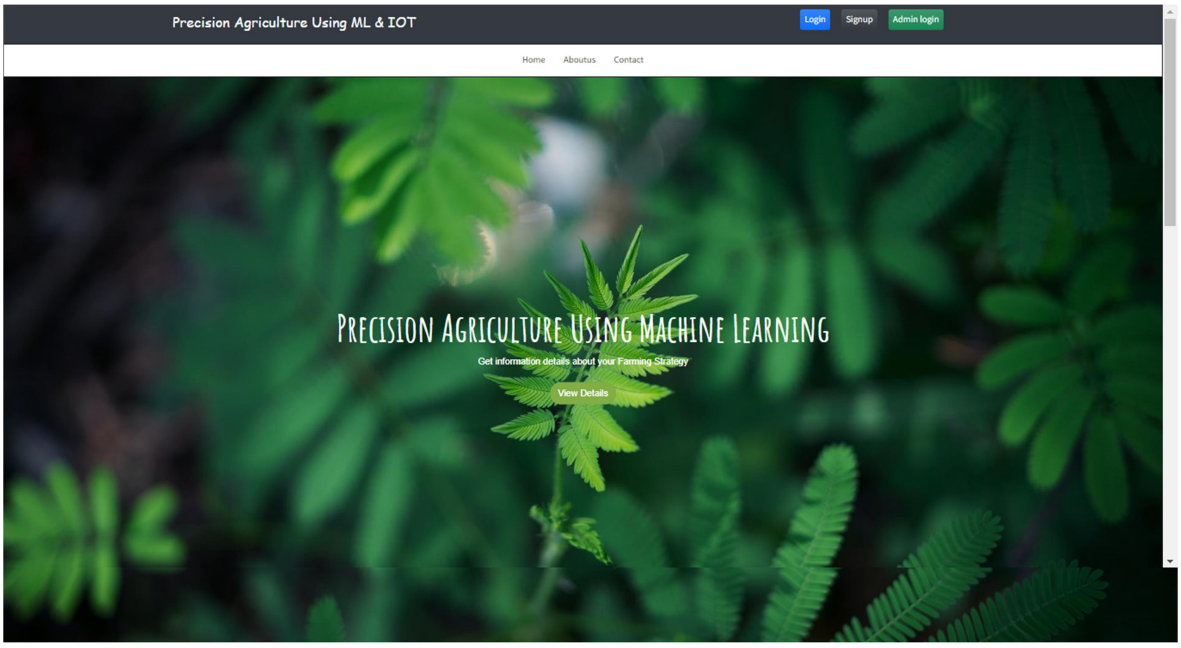
Fig-10 Output crop recommender

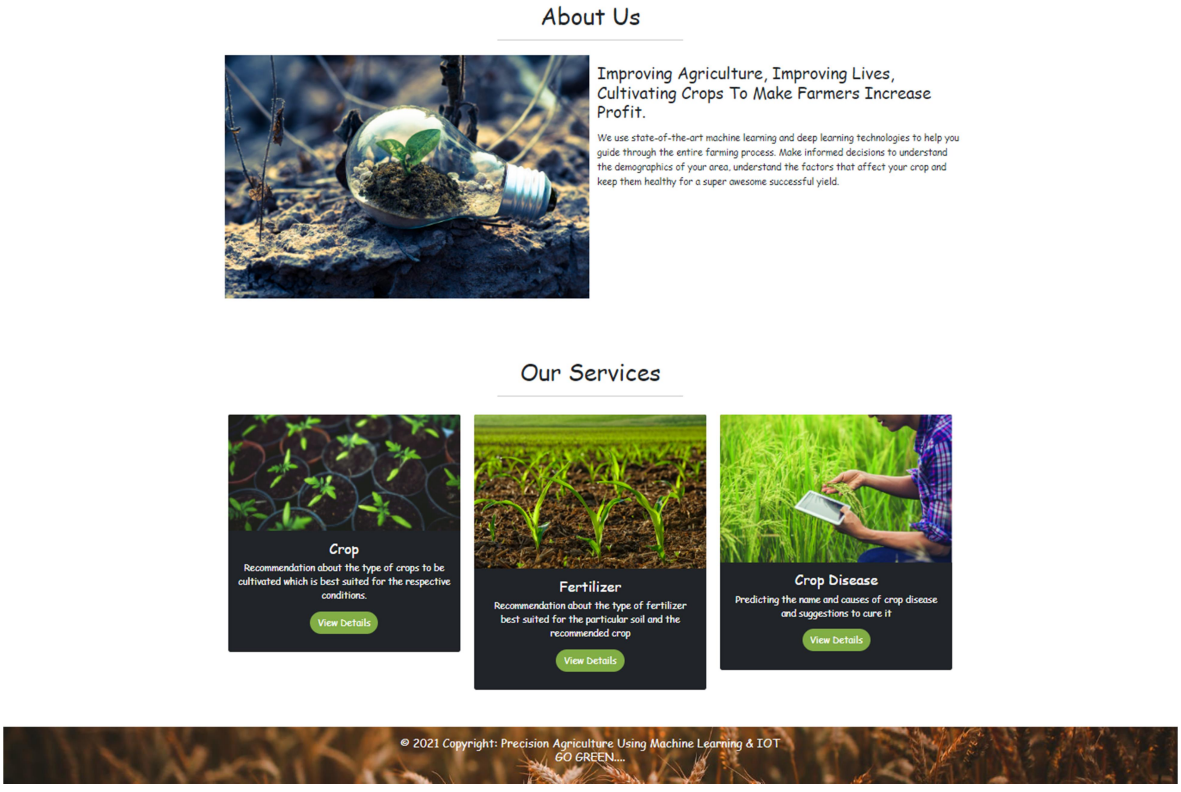
Fertilizer Recommender/Suggestion

The fertilizer recommendation system is purely python logic based. In this we compare the data (optimum nutrients for growing the crop) with the user's entered data. Then nutrient having maximum difference is made as HIGH or LOW and according to that suggestions will be fetched.

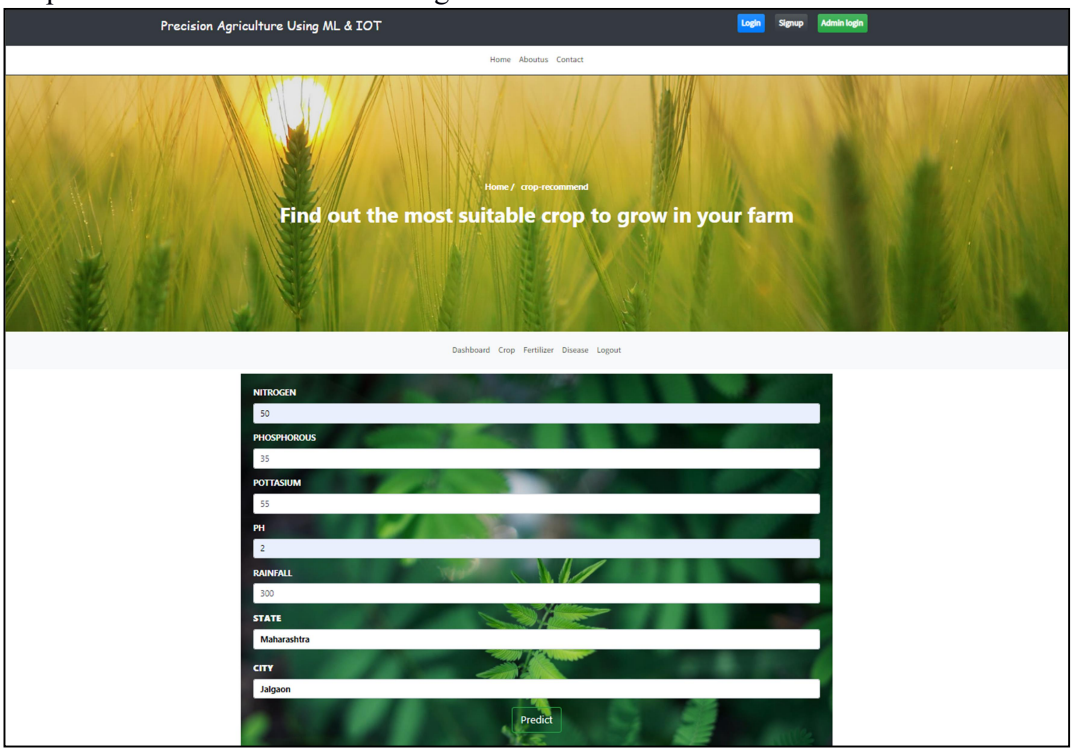
9.2 Results With Screenshots

Home Page

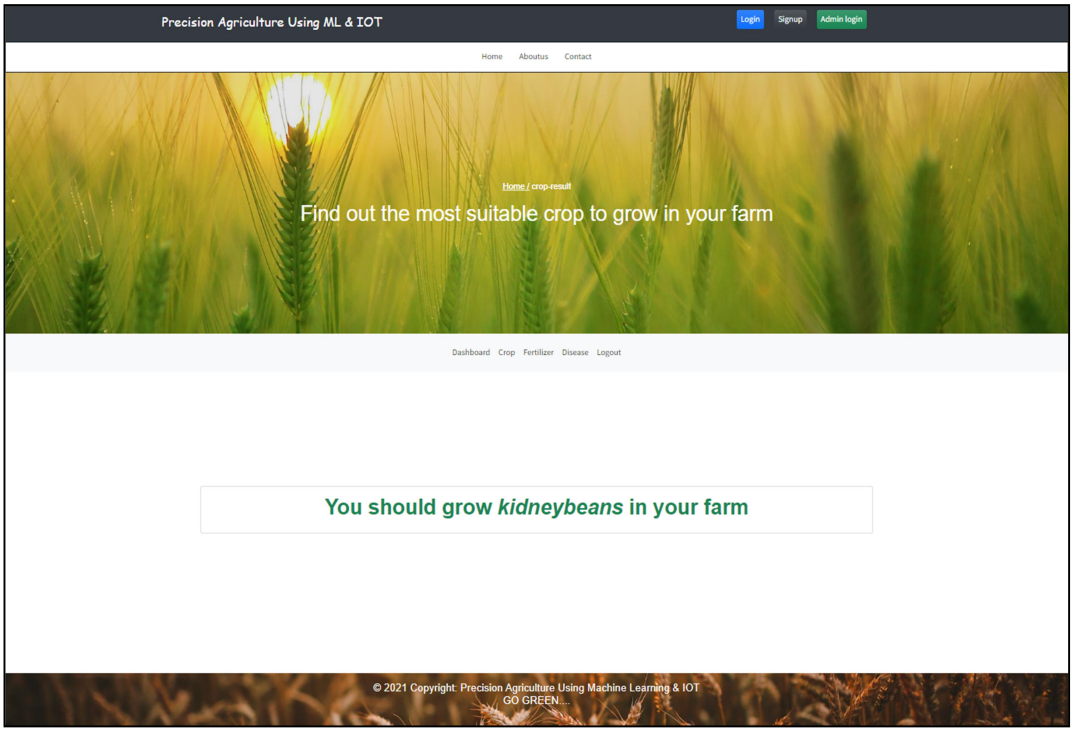




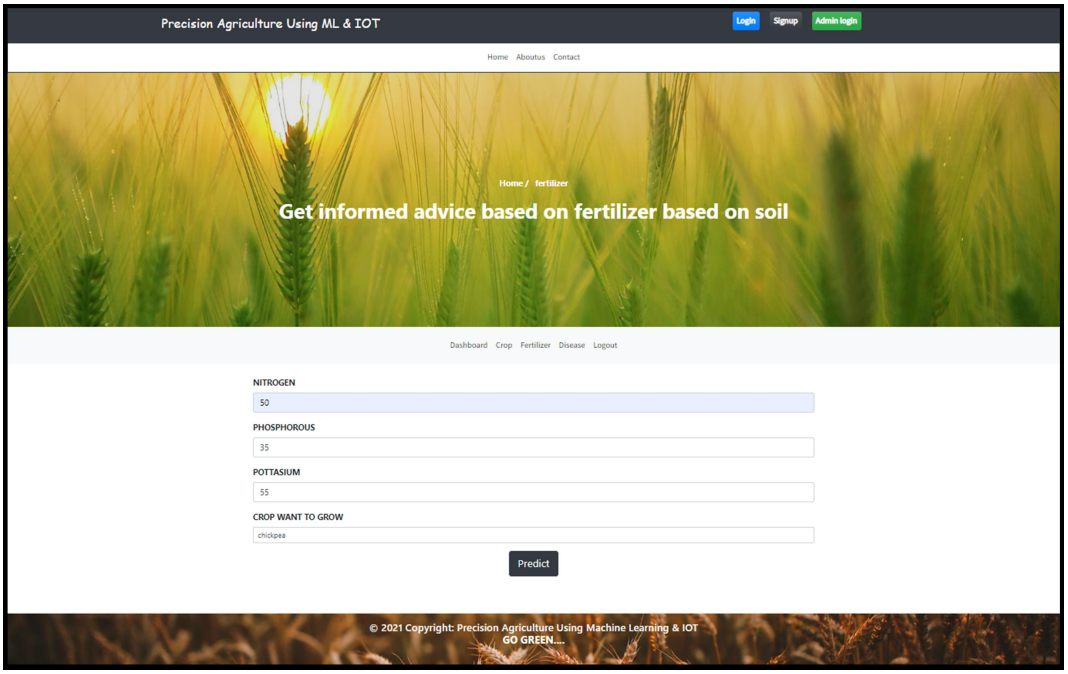
Crop Recommendation & Result Page

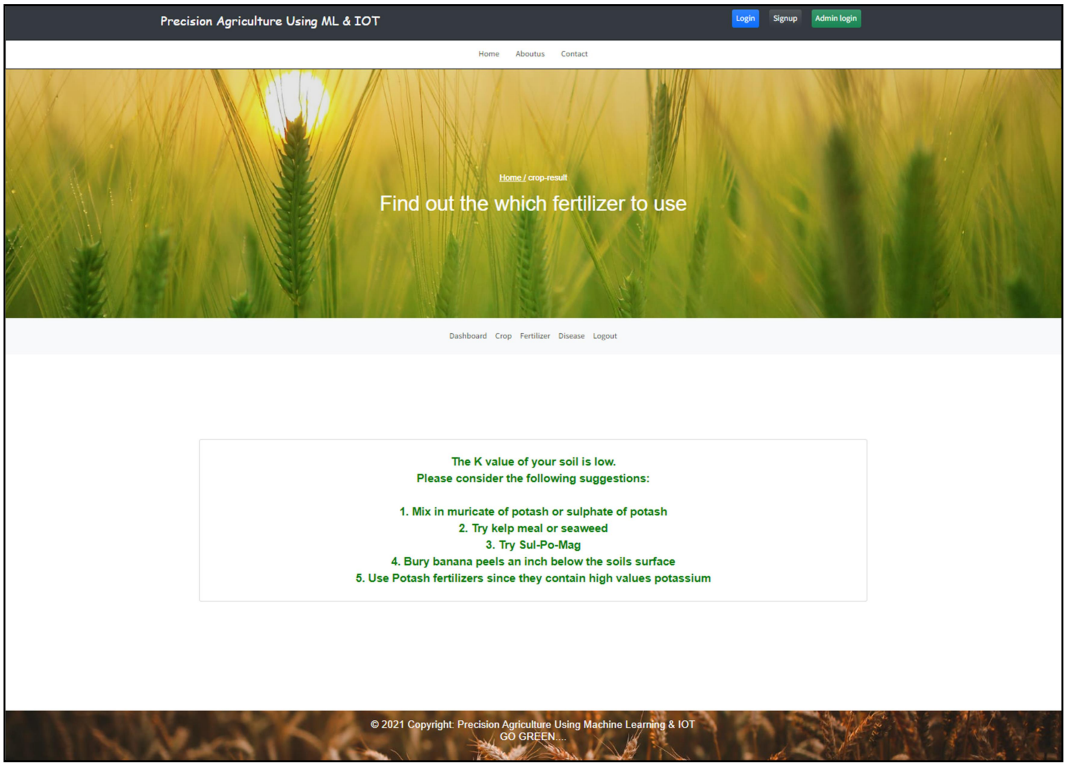


Result

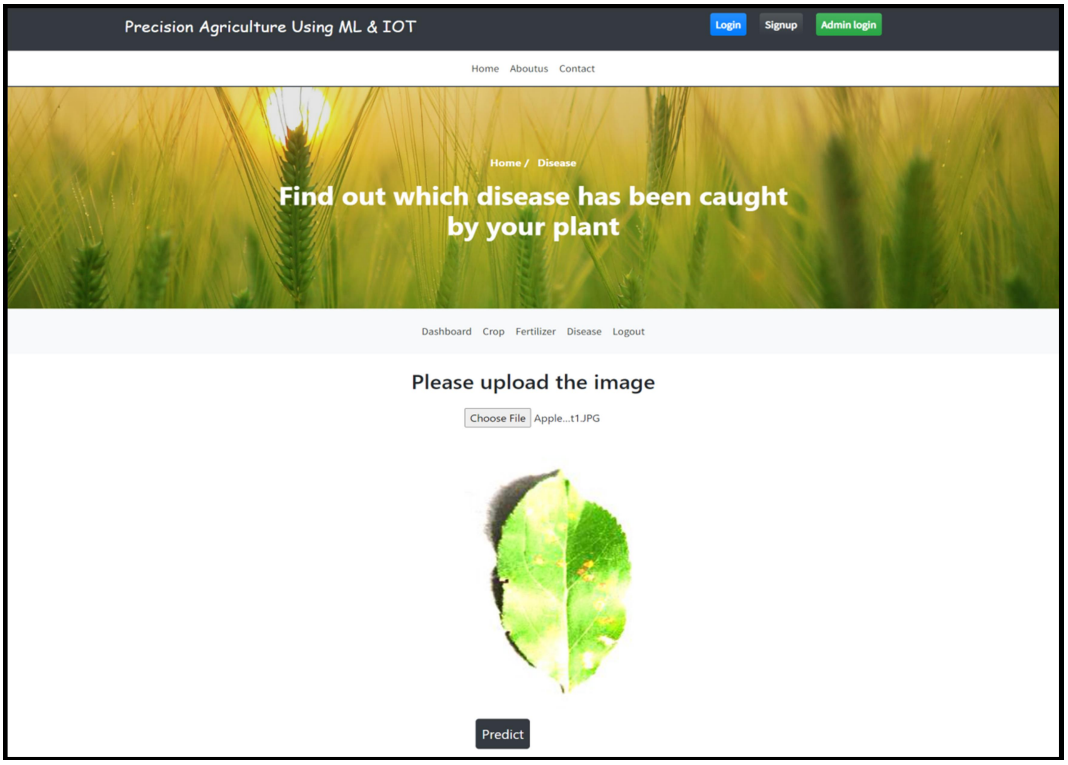


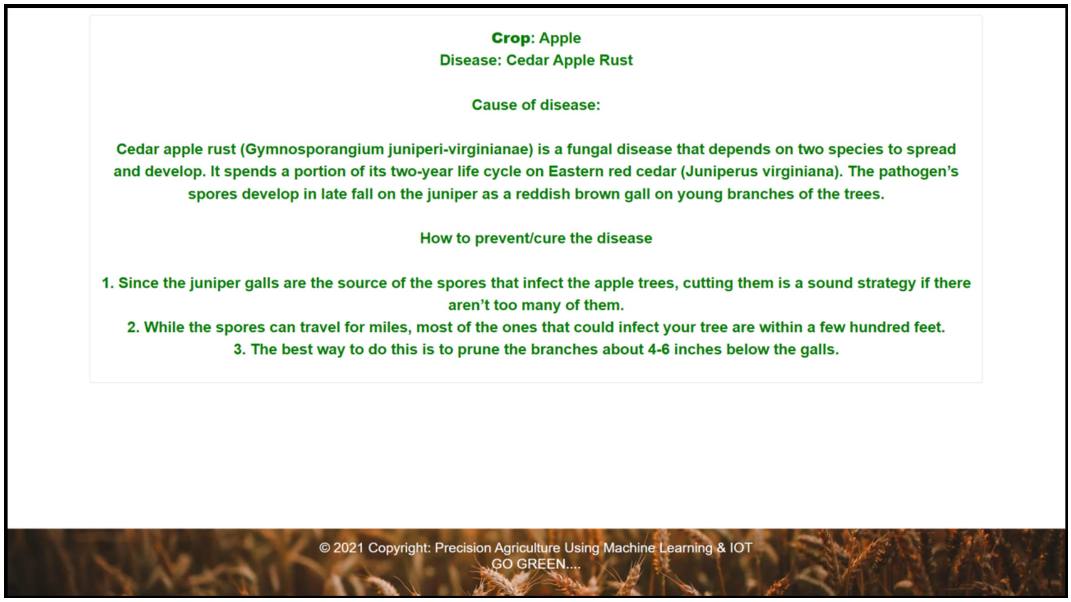
Fertilizer-Suggestion & Result Page





Crop Disease Prediction & Result Page





9.3 Advantages

- Improve farm management efficiency by adjusting field/crop treatments
- Ensure profitability, sustainability and protection of the environment
- To use new technologies to increase crop yields and profitability while lowering the levels of traditional inputs needed to grow crops
- Optimize efforts and resources, reduce consumption and waste, and boost land productivity
- It will reduce excessive chemical usage in crop production.

9.4 Limitations

- Extremely demanding work particularly collecting and then analyzing the data.
- Accuracy depends upon input dataset
- Most of the farmer not aware of such program like precision agriculture using machine learning farmer don’t know the technology
- Complexity grows with data.

CHAPTER 10

Conclusion

All This system helps the farmer to choose the right crop by providing insights that ordinary farmers don't keep track of thereby decreasing the chances of crop failure and increasing productivity. It also prevents them from incurring losses. The system can be extended to the web and can be accessed by millions of farmers across the country. We could achieve an accuracy of 90 percent from the Decision Trees, an accuracy of 70.6 percent from the Support Vector Machine, an accuracy of 94.30 percent from the Logistic Regression and an accuracy of 99.09 percent from the Random Forest model. Further development is to integrate the crop recommendation system with another subsystem, yield predictor that would also provide the farmer an estimate of production if he plants the recommended crop.

.

CHAPTER 11

References

[1] 2019, 10th International Conference on Computing, Communication and Networking Technologies, “Low-cost IOT+ML design for smart farming with multiple applications”, Fahad Kamraan Syed, Agniswar Paul, Ajay Kumar, Jaideep Cherukuri.

[2] 2019 IEEE “ Smart Management of Crop Cultivation using IoT and Machine Learning” Archana Gupta, Dharmil Nagda, Pratiksha Nikhare, Atharva Sandbhor

[3] Radhika, Narendiran, “Kind of Crops and Small Plants Prediction using IoT with Machine Learning,” International Journal of Computer & Mathematical Sciences, 2018.

[4] “Crop Recommendation on Analyzing Soil Using Machine Learning” Anguraj.Ka, Thiyaneswaran.Bb, Megashree.Gc, Preetha Shri.J.Gd, Navya.Se, Jayanthi. Jf, 2020.

[5] “Classification of Soil and Crop Suggestion using Machine Learning Techniques”, A. Mythili , IEEE 2019.

[6] Mehta, P., Shah, H., Kori, V., Vikani, V., Shukla, S., & Shenoy, M.,2018. “Survey of unsupervised machine learning algorithms on precision agricultural data”, IEEE

[7] “IOT based Crop Recommendation, Crop Disease Prediction and Its Solution” Rani Holambe, Pooja Patil, Padmaja Pawar, Saurabh Salunkhe , Mr. Hrushikesh Joshi, 2019 IRJET

[8] Tom M. Mitchell, Machine Learning, India Edition 2013, McGrawHill Education.

[9] https://data.gov.in/

[10] Kaggle: https://www.kaggle.com/datasets