1. INTRODUCTION

Agriculture has an extensive history in India. Recently, India is ranked second in the farm output worldwide [15]. Agriculture-related industries such as forestry and fisheries contributed for 16.6% of 2009 GDP and around 50% of the total workforce. Agriculture's monetary contribution to India's GDP is decreasing Farmers help to feed a nation whose population is nearly 1.4 billion, however the productivity of farms is threatened by various natural factors that ruin the crops and farmer's livelihood. Enhancing Crop Productivity is a small initiative enhancing agriculture, making smart decisions to consider the demographics of the field, the factors affecting the crop, as well as how to keep the farm healthy for a super awesome yield. This will be implemented in the form of a website providing features of Crop Recommendation, Fertilizer Recommendation and Prediction of Crop Production based on site specific parameters.

1.1 MOTIVATION

The impact of Climate Change in India. Most of the agriculture Crops are being Badly affected. Enhancing Crop Productivity is a website made for farmers to help them with crop recommendations based on values of N, P, K, temperature, rainfall, relative humidity and ph. Generally, soil gets degraded and productivity is reduced if the right crop isn't chosen, but Enhancing Crop Productivity makes it really easy by using the ML model to make the real time prediction. Second feature is Fertilizer Prediction. If the farmer opts not to change the crop as per land, he can go with the same crop but use fertilizer which will be recommended by Enhancing Crop Productivity based on N, P, K and crop values. Lastly, a very useful feature implemented is Prediction of Crop Production. Generally, it's seen that tests for soil are done by Indian government and results come within a few days but farmers really don't know much on what to do next, so Enhancing Crop Productivity is sort of their next step. A simple, intuitive website will really help farmers to easily know the whereabouts of crops, thus helping every possible bit which Enhancing Crop Productivity can. Hence, the three modules: Crop, Fertilizer, Production really comes handy and a boon for farmers.

1.2 PROBLEM DEFINITION

Enhancing Crop Productivity aims to help Indian farmers and reduce their hardship. The problems faced by Indian farmers are defined as follows:

- Productivity needs to be increased so that farmers can get more pay from the same piece of land without degrading soil.
- Indian farmers aren't able to choose the right crop based on their soil requirements depending upon factors like N, P, K, temperature, humidity, rainfall, ph.
- Farmers are generally unaware about the organic fertilizers or standard fertilizers to use as per soil requirements.
- Due to inadequate and imbalanced fertilization, soil degradation is occurring, which leads to nutrient mining and the development of second-generation problems in nutrient management.
- According to a study by the Associated Chambers of Commerce and Industry of India, annual crop losses due to pests amount to Rs. 50,000 crores.

1.3 OBJECTIVE OF PROJECT

Corresponding to problems cited above, following are objectives that "Enhancing Crop Productivity" is trying to solve:

- To implement precision agriculture (A modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection and suggests the farmers the right crop based on their site-specific parameters to reduce the wrong choice on a crop and increase in productivity).
- To solve the problem by proposing a recommendation system through an ensemble model with majority voting technique crop for the site-specific parameters with high accuracy and efficiency.
- To recommend organic fertilizer on the basis of N, P, K values and crop.
- To recognize the pest and recommend particular pesticide available in India as per ISO standards (ISO 9001, ISO 14001, ISO 17025).
- To design a web application for achieving above objectives.

1.4 LIMITATIONS

Every system cannot be perfect and have limitations to which they are confined to.

- Enhancing Crop Productivity supports 22 crops: apple, banana, black gram, chickpea, coconut, coffee, cotton, grapes, jute, kidney beans, lentil, maize, mango, moth beans, moonbeam, muskmelon, orange, papaya, pigeon peas, pomegranate, rice, watermelon Hence the user will get results which best suit the land but only from these 22 crops.
- The user must be connected to the internet so as to access the web application.
- The user must enter realistic values for getting the best result. (Though the invalid values are not accepted).
- The maximum file size in case of image upload is 2 GB and maximum dimensions as per Web format are: 16383 x 16383.

2. LITERATURE SURVEY

Agriculture is a major source of livelihood in India and Indian farmers put in their heart and soul to feed people. Farmers generally deal with crops, fertilizers, pests and pesticides. Hence, Enhancing Crop Productivity aims to serve Indian farmers via all three modules of Crop Recommendation, Fertilizer Recommendation and Pesticide Recommendation. Crop recommendation has been an area which is explored a lot, but all of the systems vary on the basis of parameters that are fed into the ML model. Most of the ML models use Random Forest, some use Decision Tree, while others use Ensemble methods via Majority Voting Mechanism. Fertilizer Recommendation doesn't work much in the area of AI. Main reason can be disintegrated data, but Enhancing Crop Productivity collected all of the data from various sources and integrated it to have a well-formed dataset. A dictionary-based solution is implemented in Enhancing Crop Productivity. Thirdly, Pesticide Recommendation is not at all touched area, researchers have just restricted it to Pest Detection only, but Enhancing Crop Productivity extends the idea of identification of pest, along with a dictionary-based solution for the corresponding pesticide, available in India. Enhancing Crop Productivity uses ISO 9001, ISO 14001 and ISO 17025 standards for pesticide recommendation. Most of the pesticides are taken from biostudy site which is a really popular site for farmers but the problem is that search isn’t easy there and maximum pesticides recommended aren’t available in India. Following is discussed about various research papers pertaining to services offered by Enhancing Crop Productivity.

2.1 INTRODUCTION

The main purpose of the literature review work was to survey previous studies on knowledge sharing and intranets. In this, we look into the details about the existing system and try to reduce the disadvantages of the existing system. We try to improve the performance and the efficiency of the new proposed system and also learn the advantages of proposed system.

***Table2.1**: Analysis of literature*

SNO	YEAR	NAME OF THE PAPER	ALGORITHM USED	DRAWBACKS
1	2015	Analysis of Soil Behavior and Prediction of Crop Yield Using Data Mining Approach	Naïve Bayes, k-nearest neighbour	Choosing K is tricky. Conditional independence assumption is violated by real-world data.
2	2016	Rice crop yield prediction in India using support vector machines	Support vector machine	Requires more time to process. SVM requires a long training period
3	2018	Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge	Gradient boosting tree, linear regression	slow to train and it also requires a large amount of memory, especially when the number of trees is large.
4	2019	Paddy acreage mapping and yield prediction using sentinel-based optical and SAR data in Sahib Ganj district, Jharkhand (India)	Linear regression	Prone to underfitting Sensitive to outliers

2.2 EXISTING SYSTEM

A country relies on agriculture for its economic development. In this case, the rate of crop production plays a major role in the country's economic development. So, a need to raise crop production rates. In the past the crop, fertilizer and yield prediction were done by experienced farmers. Then data mining techniques were used for prediction. Training the model with decision trees.

Disadvantages

- Slight change in data may cause a significant change in result.
- With large data, the prediction stage might be slow.
- Overfitting problem
- For a Decision tree sometimes, calculation can go far more complex compared to other algorithms.
- Decision tree often involves higher time to train the model.

2.3 PROPOSED SYSTEM

The proposed system concentrates on prediction of crop, fertilizer and production using the Random Forest algorithm based on existing data. Random Forest Algorithm is a machine learning algorithm which belongs to the technique of supervised learning. The main objective is to find information of climate parameters temperature, humidity, soil quality, etc. in the dataset containing output details for the past 12 years.

Advantages

- Adapts quickly to the dataset.
- It can perform both regression and classification tasks.
- A random forest produces good predictions that can be understood easily.
- The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

3. SYSTEM ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

3.1 INTRODUCTION

In this phase the requirements are gathered and analysed. User's requirements are gathered in this phase. This phase is the main focus of the users and their interaction with the system.

There are few questions raised:

- Who is going to use the system?
- How will they use the system?
- What data should be input into the system?
- What data should be output by the system?

These general questions are answered during a requirement gathering phase. After requirement gathering these requirements are analysed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3.2 SOFTWARE REQUIREMENTS

It deals with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements are description of features and functionalities of the target system.

Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

Functional Requirements

It has three different modules namely: Crop Recommendation, Fertilizer Recommendation and Pesticide Recommendation.

So, this section will define functional requirements for all the modules separately.

- **Crop Recommendation**: The system will recommend the crop as per site specific parameters entered by the user.
- **Fertilizer Recommendation**: The system will recommend the organic fertilizers as per the values entered by the user.
- **Crop Production Prediction**: The system will predict the crop prediction as per the values entered by the user.

Software Requirements

For developing the application, the following are the Software Requirements:

- **Operating System:** Windows 10
- **Coding Language:** Python 3.11.3
- **Web Framework:** Flask
- **IDE:** PyCharm

3.3 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, a hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

The hardware requirements that are required for this project are as follows:

- **System:** Pentium i3 Processor.
- **Hard Disk:** 500 GB
- **RAM:** 4 GB

3.4 WORKFLOW DIAGRAM OF PROJECT

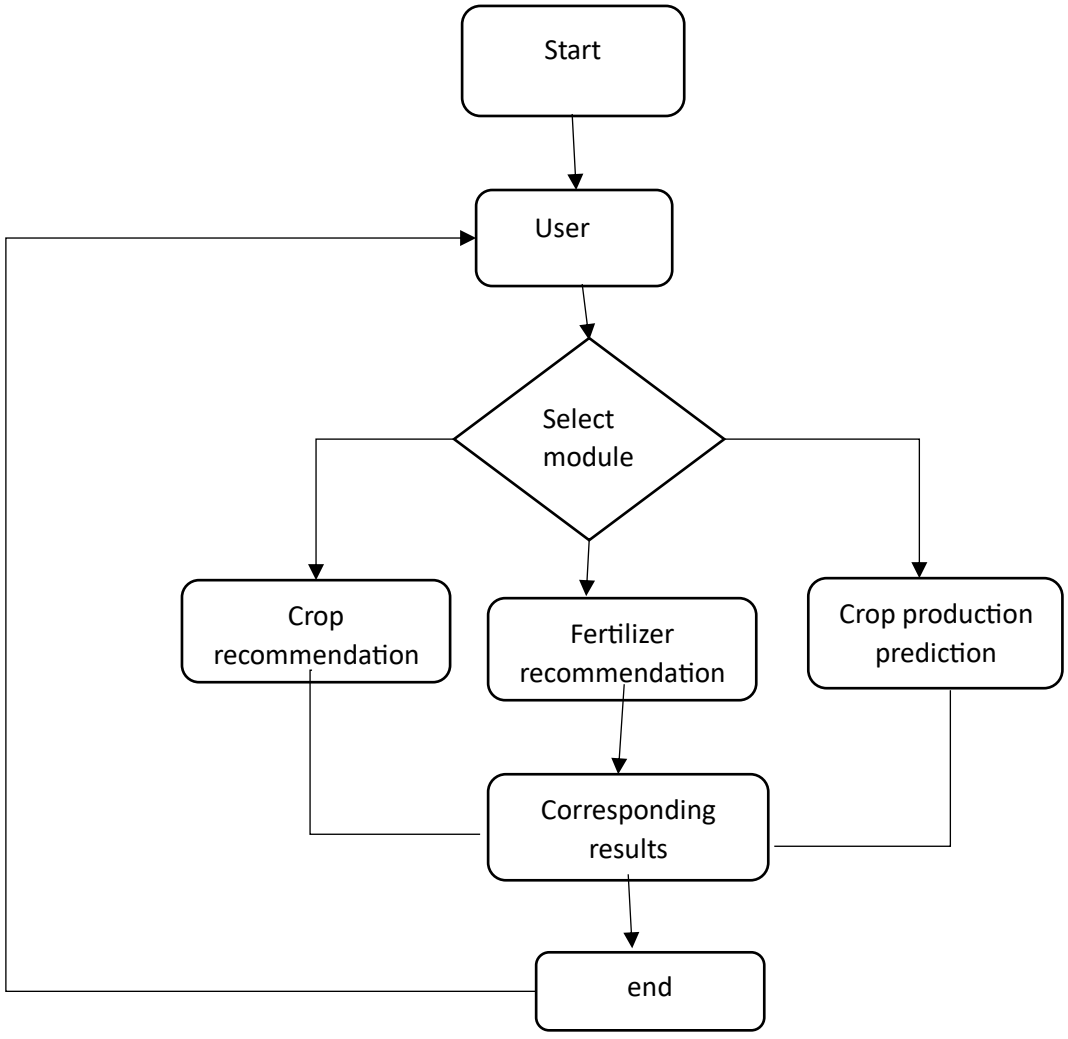

***Figure3.1**: Workflow diagram of the project*

4. SYSTEM DESIGN

The process of design involves “conceiving and planning out in mind and making a drawing, pattern or a sketch”. The system design transforms a logical representation of what a given system is required to do into the physical reality during development. Important design factors such as reliability, response time, throughput of the system, maintainability, expandability etc., should be taken into account. Design constraints like cost, hardware limitations, standard compliance etc should also be dealt with. The task of system design is to take the description and associate with it a specific set of facilities-men, machines (computing and other), accommodation, etc., to provide complete specifications of a workable system. This new system must provide for all of the essential data processing and it may also do some of those tasks identified during the work of analysis as optional extras. It must work within the imposed constraints and show improvement over the existing system. At the outset of design, a choice must be made between the main approaches.

Talks of ‘preliminary design’ concerned with identification analysis and selections of the major design options are available for development and implementation of a system. These options are most readily distinguished in terms of the physical facilities to be used for the processing who or what does the work.

4.1 INTRODUCTION

Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming, as in a stylized software engineering process." Software design usually involves problem solving and planning a software solution. This includes both a low-level component design and a high level, architecture design.

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analysed and specified the software design involves four technical activities – design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system.

Design is the only way to accurately translate the customer's requirements into finished software or a system.

4.2 DFD/ER/UML DIAGRAMS

UML stands for Unified Modelling Language which is used in object-oriented software engineering. It is a standard language for specifying, visualizing, constructing, and documenting the artefacts of the software systems. UML is different from other common programming languages like C++, Java, and COBOL etc. It is pictorial language used to make software blueprints.

Although typically used in software engineering it is a rich language that can be used to model an application structure, behaviour and even business processes. There are 8 UML diagram types to help us model this behaviour.

There are two types of UML modelling:

- Structural Modelling
- Behavioural Modelling

Structural Modelling:

Structural model represents the framework for the system and this framework is the place where all other components exist. Hence, the class diagram, component diagram and deployment diagrams are part of structural modelling. They all represent the elements and the mechanism to assemble them.

The structural model never describes the dynamic behaviour of the system. Class diagram is the most widely used structural diagram.

Structural Modelling captures the static features of a system. They consist of the following:

i. Classes diagrams
ii. Objects diagrams
iii. Deployment diagrams
iv. Package diagrams
v. Composite structure diagram
vi. Component diagram

Behavioural Modelling:

Behavioural model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioural modelling shows the dynamic nature of the system.

They consist of the following:

i. Activity diagrams
ii. Interaction diagrams
iii. Use case diagrams

All the above show the dynamic sequence of flow in a system.

4.2.1 USE CASE DIAGRAM

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. The "actors" are people or entities operating under defined roles within the system. Here the actors are user and system.

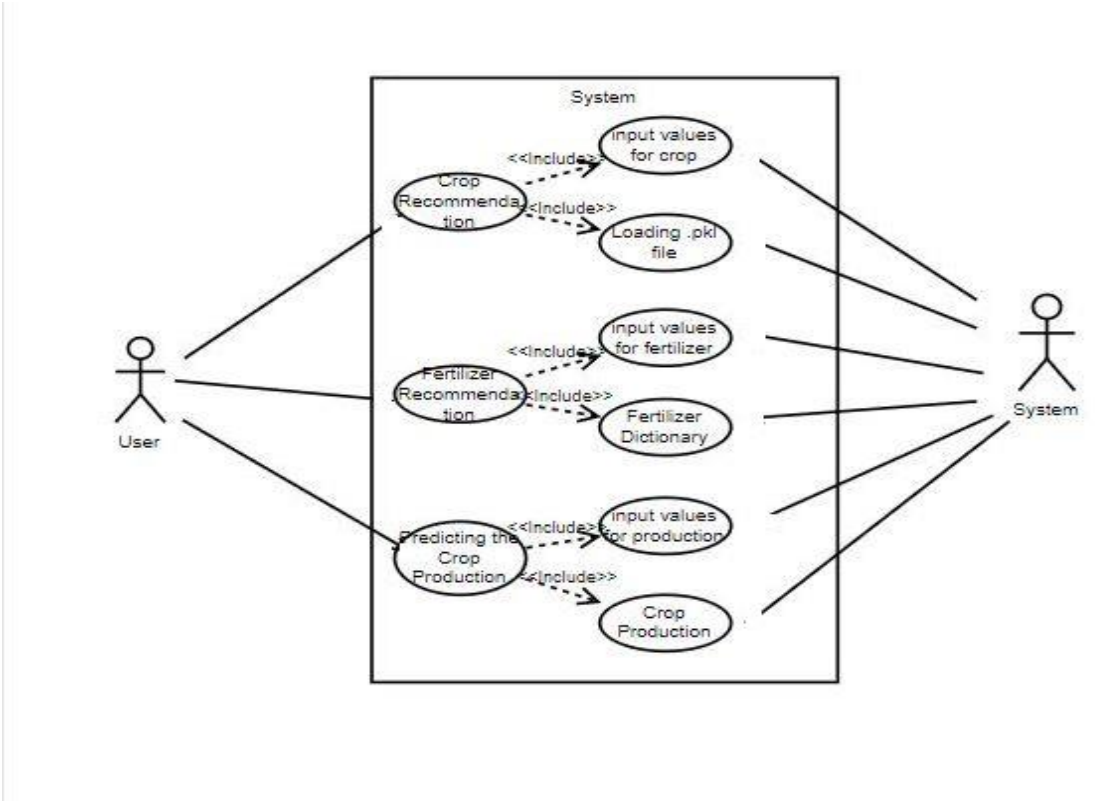

***Figure 4.1**: Use case diagram*

The above use case diagram describes each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

4.2.2 CLASS DIAGRAM

Class diagrams are the main building blocks of every object-oriented method. It is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application The project consists of four classes named user, crop recommendation, fertilizer recommendation and crop production prediction. Firstly, the user will go to home page and he will select the module of his choice and enter the parameters and the system will recommend the crop, fertilizer and production of the crop as per the parameters given by the users.

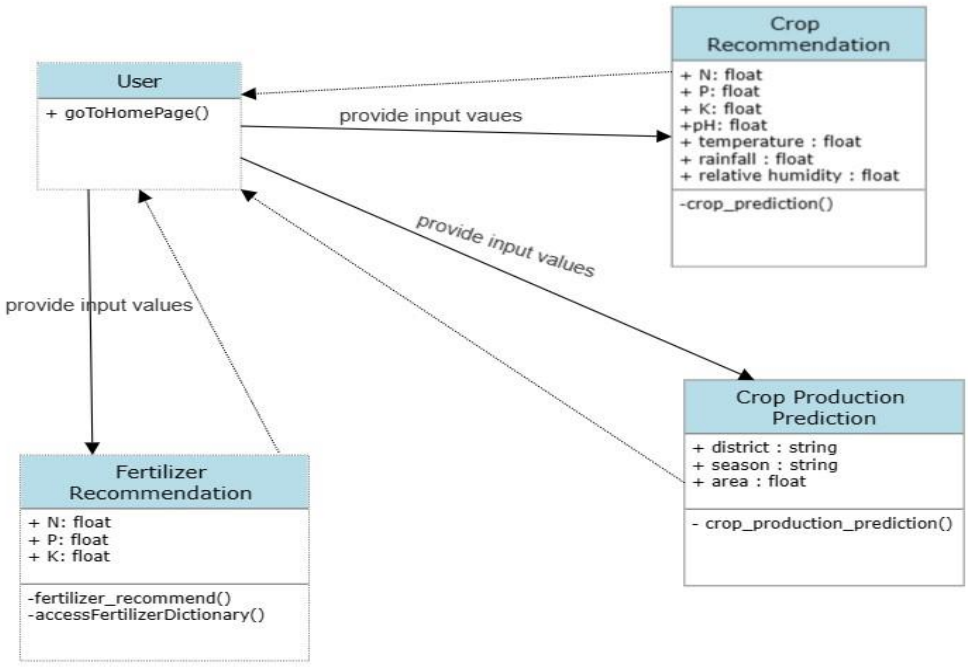

Figure 4.2: *Class diagram*

The above class diagram has four classes consists of user, crop recommendation, fertilizer recommendation and crop production prediction. Each class has its own operations and attributes.

4.2.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems. nature but from a different angle.

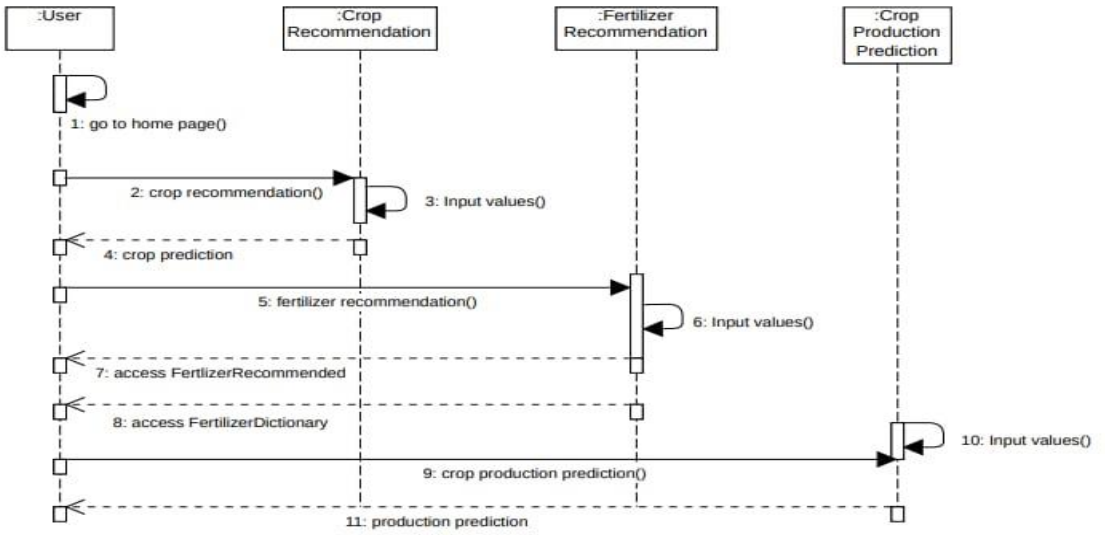

Figure 4.3: *Sequence diagram*

The above sequence diagram has four objects namely user, crop recommendation, fertilizer recommendation and crop production prediction. Which will represent the flow of messages in the system.

4.2.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

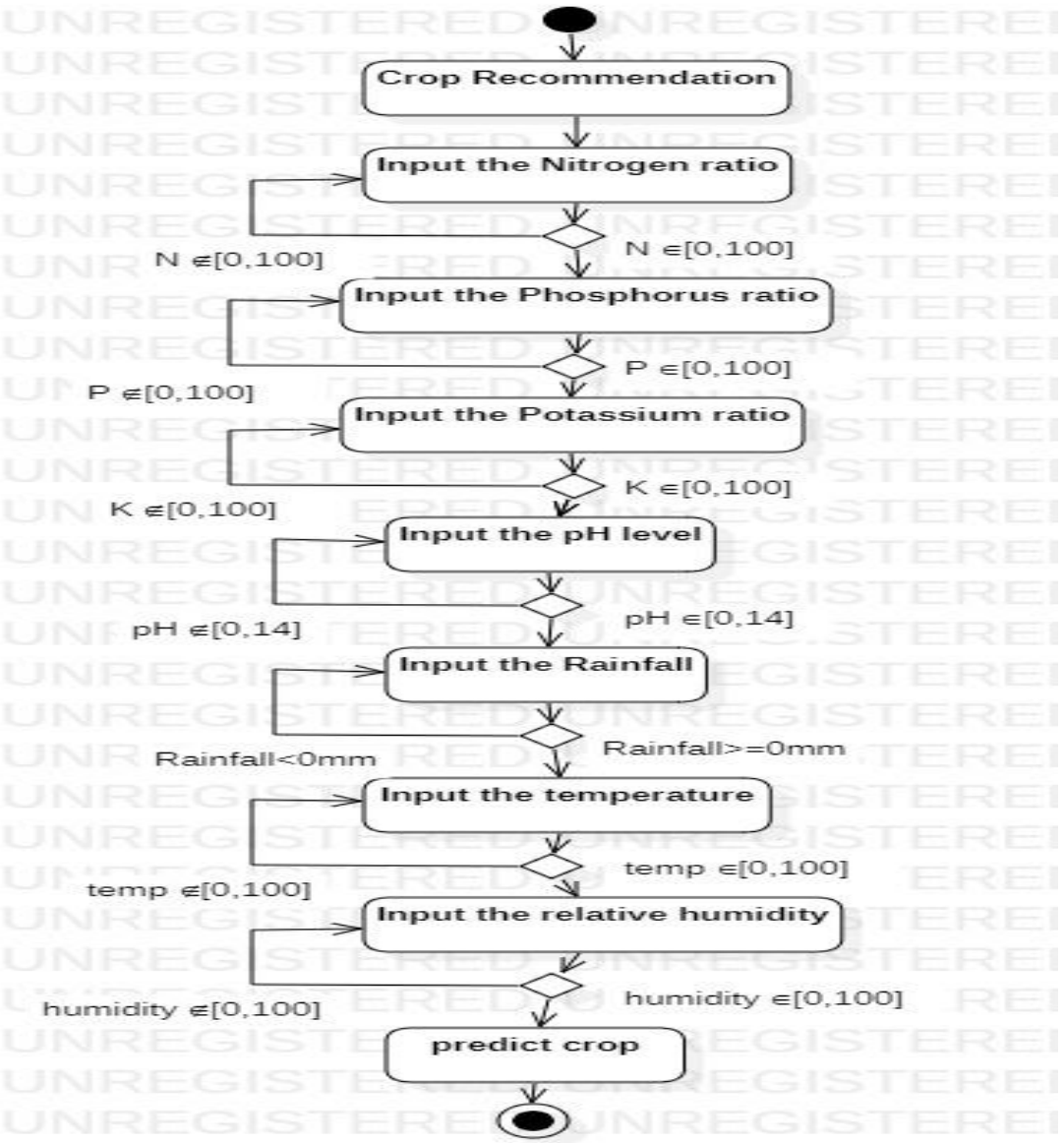

Figure 4.4: *Activity diagram for crop recommendation*

The above activity diagram describes about crop recommendation module, here the user needs to give input values like N, P, K, pH, rainfall, temperature, and humidity so that system will recommend the crop to the user

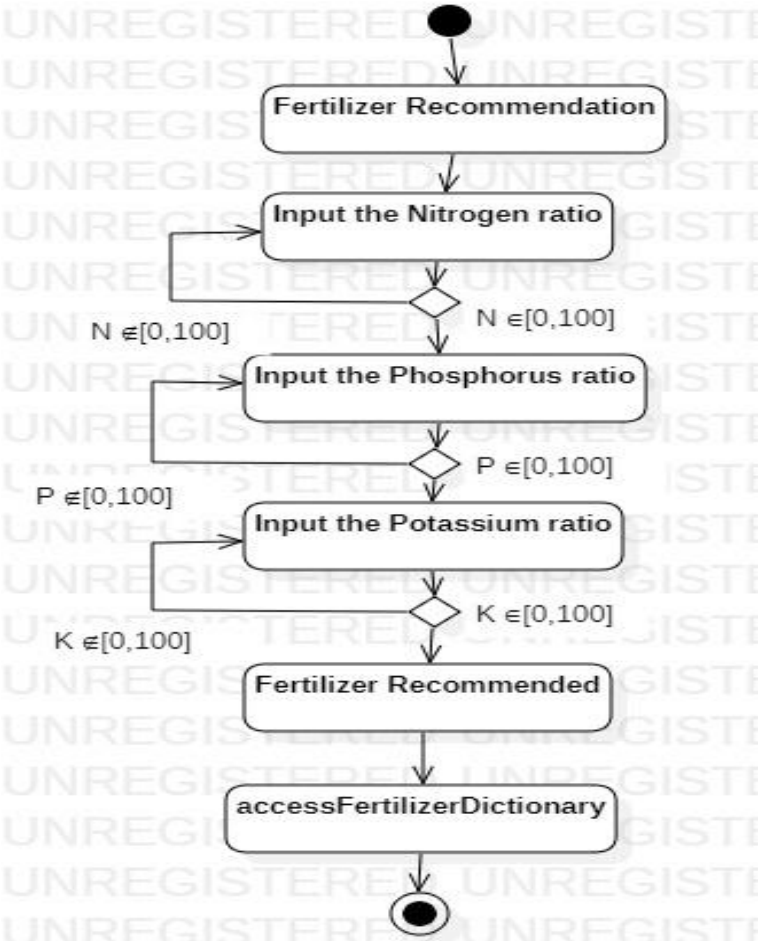

Figure 4.5: *Activity diagram for fertilizer recommendation*

The above activity diagram describes about fertilizer recommendation module, here the user needs to give input values like N, P, K values based on the dictionary-based suggestions it will recommend the fertilizer to the user

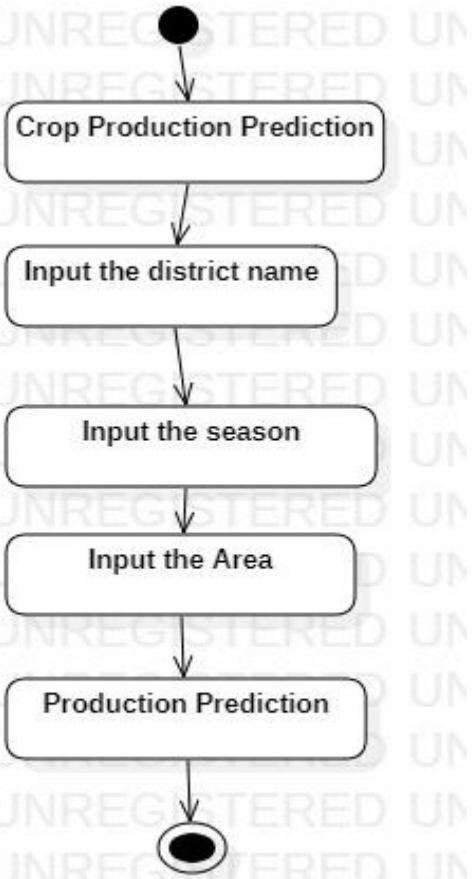

Figure 4.6: *Activity diagram for crop production prediction*

The above activity diagram describes about crop production prediction module, here the user needs to give input values like district name, season and area so that system will predict the crop production to the user.

5. IMPLEMENTATION AND RESULTS

5.1 INTRODUCTION

Functions are used for placing or storing the code which is to be repeated several times. For example, if we need same code, then we must have to write that code again and again. So in order to remove this we use functions.

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system is giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over an evaluation of change over methods apart from planning. Two major tasks for preparing the implementation are education and training of the users and testing of the system.

5.2 METHOD OF IMPLEMENTATION

- **Crop Recommendation**

  This module can be implemented in four steps

  Step 1: Data Acquisition Dataset can be acquired from Kaggle. Click here to have a look at the dataset.

  Step 2: Values Input Users are expected to input the site-specific parameters like: N, P, K (all of them in %), temperature (in °C), relative humidity (in %), rainfall (in mm) and ph.

  Step 3: ML Model Training and creating .pkl file Recommendation system is based on the ensemble model with majority voting technique. The constituent models are:

  1. SVM
  2. Random Forest
  3. Naive Bayes

4. KNN

After the model is trained, a .pkl file is created.

Step 4: Crop Recommendation .pkl file is loaded to recommend the crop based on input.

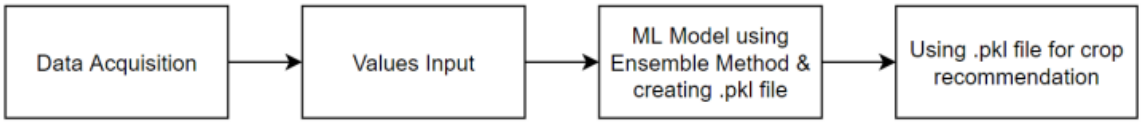

Figure 5.2.1*: *Methodology for Crop Recommendation

- **Fertilizer Recommendation**

  This module can be implemented in four steps

  Step 1: Data Acquisition Dataset will be created manually after collecting data from verified sources listed below:

  1. The Fertilizer Association of India
  2. Indian Institute of Water Management
  3. Kaggle The columns of the dataset are: N, P, K (all of them in %) and crop.

  Step 2: Values Input Users are expected to input the site-specific parameters like: N, P, K (all of them in %), and crop (select from list - only 22 crops supported).

  Step 3: Difference between desired and actual Difference is calculated between desired value of N, P, K as per crop and the farm's actual value, based on it there are 3 outcomes possible for all three nutrients:

  1. High
  2. Low
  3. Up to the mark

  Step 4: Fertilizer Recommendation Based on the outcomes from the above step, a dictionary-based solution (organic fertilizers) will be displayed.

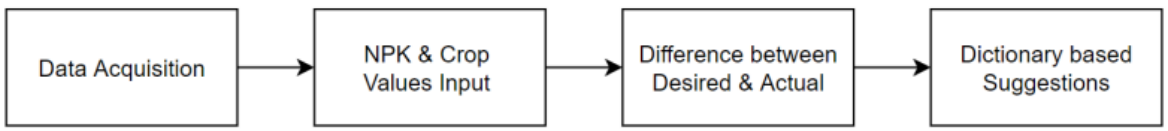

***Figure 5.2.2:** Methodology for Fertilizer Recommendation*

- **Crop Production Prediction**

  Step 1: Data Acquisition Dataset will be created manually after collecting data from verified source listed below:

  Kaggle The columns of the dataset are: state name, district name, year, season, crop, area, production

  Step 2: Values Input Users are expected to input the site-specific parameters like: district, season, area.

  Step 3: ML Model will predict the crop production.

  Step 4: Based on given parameters the system will predict the production of the crop.

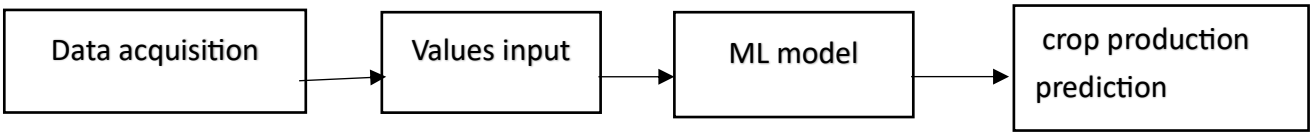

Figure5.2.3: *Methodology for crop production prediction*

5.2.1 TECHNOLOGIES

The technologies that are used in the project are:

- Machine Learning
- Python
- NumPy
- Pandas
- Flask
- Pickle
- PyCharm
- TensorFlow
- Sklearn

1. Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: ***The ability to learn***. Machine learning is actively being used today, perhaps in many more places than one would expect.

Classification of Machine Learning:

Machine learning implementations are classified into three major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:

- **Supervised learning:**

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

- **Unsupervised learning:**

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own comes under the category of Unsupervised Learning. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values.

They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms. As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

- **Reinforcement learning:**

When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences.

In the human world, it is just like learning by trial and error. Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others.

- **Semi-supervised learning:**

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

2. Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

3. NumPy

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

4. Pandas

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and

intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

5. **Flask**

   Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. Lightweight.

6. **Pickle**

   Pickling" is the process whereby a Python object hierarchy is converted into a byte stream, Pickle is used to serialize and deserialize a python object structure. Any object on python can be pickled so that it can be saved on disk.

7. **PyCharm**

   PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development

8. **TensorFlow**

   TensorFlow is an end-to-end open-source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

9. **Sklearn**

   Scikit-learn is an open-source data analysis library, and the gold standard for Machine Learning (ML) in the Python ecosystem. Key concepts and features include: Algorithmic decision-making methods, including: Classification: identifying and categorizing data based on patterns.

5.3 EXPLANATIONS OF KEY FUNCTIONS

Source Code:

Crop Recommendation

```
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn import model_selection

crop = pd.read_csv('Data/crop_recommendation.csv')
X = crop.iloc[:,:-1].values
Y = crop.iloc[:,-1].values

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.15)

models = []
models.append(('SVC', SVC(gamma ='auto', probability = True)))
models.append(('svm1', SVC(probability=True, kernel='poly', degree=1)))
models.append(('svm2', SVC(probability=True, kernel='poly', degree=2)))
models.append(('svm3', SVC(probability=True, kernel='poly', degree=3)))
models.append(('svm4', SVC(probability=True, kernel='poly', degree=4)))
models.append(('svm5', SVC(probability=True, kernel='poly', degree=5)))
models.append(('rf',RandomForestClassifier(n_estimators = 21)))
models.append(('gnb',GaussianNB()))
models.append(('knn1', KNeighborsClassifier(n_neighbors=1)))
models.append(('knn3', KNeighborsClassifier(n_neighbors=3)))
models.append(('knn5', KNeighborsClassifier(n_neighbors=5)))
models.append(('knn7', KNeighborsClassifier(n_neighbors=7)))
models.append(('knn9', KNeighborsClassifier(n_neighbors=9)))
```

```
vot_soft = VotingClassifier(estimators=models, voting='soft')
vot_soft.fit(X_train, y_train)
y_pred = vot_soft.predict(X_test)

scores = model_selection.cross_val_score(vot_soft, X_test, y_test,cv=5,scoring='accuracy')
print("Accuracy: ",scores.mean())

score = accuracy_score(y_test, y_pred)
print("Voting Score % d" % score)

import pickle
pkl_filename = 'Crop_Recommendation.pkl'
Model_pkl = open(pkl_filename, 'wb')
pickle.dump(vot_soft, Model_pkl)
Model_pkl.close()
```

Fertilizer Recommendation

```
fertilizer_dict = {

    'NHigh': """<b style = "color:#c79c60;">The N value of soil is high and might give rise to weeds.</b>

    <br/><br/> Please consider the following suggestions:

    <p align="justify">1. <i> Manure </i> – 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.</p>

    <p align="justify">2. <i>Coffee grinds </i> – 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.</p>

    <p align="justify">3. <i>Plant nitrogen fixing plants</i> – planting vegetables that are in Fabaceae family like peas, beans and soybeans have the ability to increase nitrogen in your soil</p>
```

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 <hr style = "height:2px; background-color:#c79c60;">""",

'Nlow': """The N value of your soil is low.

 Please consider the following suggestions:

 1. Add sawdust or fine woodchips to your soil – the carbon in the sawdust/woodchips love nitrogen and will help absorb and soak up and excess nitrogen.

2. Plant heavy nitrogen feeding plants – tomatoes, corn, broccoli, cabbage and spinach are examples of plants that thrive off nitrogen and will suck the nitrogen dry.

3. Water – soaking your soil with water will help leach the nitrogen deeper into your soil, effectively leaving less for your plants to use.

4. Sugar – In limited studies, it was shown that adding sugar to your soil can help potentially reduce the amount of nitrogen is your soil. Sugar is partially composed of carbon, an element which attracts and soaks up the nitrogen in the soil. This is similar concept to adding sawdust/woodchips which are high in carbon content.

5. Add composted manure to the soil.

6. Plant Nitrogen fixing plants like peas or beans.

7. Use NPK fertilizers with high N value.

8. Do nothing – It may seem counter-intuitive, but if you already have plants that are producing lots of foliage, it may be best to let them continue to absorb all the nitrogen to amend the soil for your next crops. <hr style = "height:2px; background-color:#c79c60;">""",

'NNo': """The N value of your soil is up to the mark.<hr style = "height:2px; background-color:#c79c60;">""",

'PHigh': """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 <hr style = "height:2px; background-color:#c79c60;">""",

'Plow': """The P value of your soil is low.

Please consider the following suggestions:

1. Bone meal – a fast acting source that is made from ground animal bones which is rich in phosphorous.

2. Rock phosphate – a slower acting source where the soil needs to convert the rock phosphate into phosphorous that the plants can use.

3. Phosphorus Fertilizers – applying a fertilizer with a high phosphorous content in the NPK ratio (example: 10-20-10, 20 being phosphorous percentage).

4. Organic compost – adding quality organic compost to your soil will help increase phosphorous content.

5. Manure – as with compost, manure can be an excellent source of phosphorous for your plants.

6. Clay soil – introducing clay particles into your soil can help retain & fix phosphorus deficiencies.

7. Ensure proper soil pH – having a pH in the 6.0 to 7.0 range has been scientifically proven to have the optimal phosphorus uptake in plants.

8. If soil pH is low, add lime or potassium carbonate to the soil as fertilizers. Pure calcium carbonate is very effective in increasing the pH value of the soil.

9. If pH is high, addition of appreciable amount of organic matter will help acidify the soil. Application of acidifying fertilizers, such as ammonium sulfate, can help lower soil pH <hr style = "height:2px; background-color:#c79c60;">""",

'PNo': """The P value of your soil is up to the mark.<hr style = "height:2px; background-color:#c79c60;">""",

'KHigh': """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.

<hr style = "height:2px; background-color:#c79c60;">""",

'Klow': """The K value of your soil is low.

Please consider the following suggestions:

1. Mix in muricate of potash or sulphate of potash

2. Try kelp meal or seaweed

3. Try Sul-Po-Mag

4. Bury banana peels an inch below the soils surface

5. Use Potash fertilizers since they contain high values potassium <hr style = "height:2px; background-color:#c79c60;">

""",

'KNo': """The K value of your soil is up to the mark.<hr style = "height:2px; background-color:#c79c60;"s>"""

}

Crop Production Prediction

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from PIL import Image
from sklearn.ensemble import RandomForestRegressor
plt.rc("font", size=14)
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
df = pd.read_csv("Data/crop_production/Crop_Production_with_rainfall.csv")
rain = pd.read_csv("Data/crop_production/Rainfall Predicted.csv")
data = df.dropna()
print(data.shape)
dis=input("Enter the District Name: ")
state=list(data[data["District_Name"]==dis.upper()]["State_Name"][:1])[0]
season=input("Enter the Season: ")
area_in=input("Enter Area in hectares: ")
s=list(data["Season"].unique())
for x in s:
    if season.title() in x:
        sin=s.index(x)
data_cu=data[data["District_Name"]==dis.upper()][data["Season"]==s[sin]]
```

```
data1 = data_cu.drop(["State_Name","Crop_Year"],axis=1)
data_dum = pd.get_dummies(data1)
x = data_dum.drop("Production",axis=1)
y = data_dum[["Production"]]

model = RandomForestRegressor()
model.fit(x,y.values.ravel())

ch=pd.DataFrame()
for crop in list(data_cu["Crop"].unique()):
    t=(x[x["Crop_{}".format(crop)]==1])[:1]
    ch=pd.concat([ch,t])
ch["Area"]=area_in
ch["Rainfall"]=list(rain[rain["State_Name"]==state]["Rainfall"])[0]
predict=model.predict(ch)
crname=data.loc[ch.index]["Crop"]
crdata= {'Crop': list(crname),
        'Production': list(predict)}
crpro = pd.DataFrame(crdata)
crpro=crpro.sort_values(by=['Production'], ascending=False)
print(crpro)
fig=plt.figure()
ax = fig.add_axes([0,0,1,1])
tstr='Predicted Production in        District: '+dis.title()+'      Season: '+season.title()
```

```
ax.set_title(tstr,fontsize=15)
ax.set_ylabel('Production in Tones', fontsize=14)
ax.set_xlabel('Crop', fontsize=13)
ax.bar(list(crpro["Crop"])[:5], list(crpro["Production"])[:5])
plt.savefig('plot.png', dpi=400, bbox_inches='tight')
im = Image.open('plot.png').show()
```

Final Code

```
from flask import Flask, render_template, request, Markup
import pandas as pd
from utils.fertilizer import fertilizer_dict
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
import pickle
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from PIL import Image
from sklearn.ensemble import RandomForestRegressor
def finding(d,se,ar):
```

```
df = pd.read_csv("Data/crop_production/Crop_Production_with_rainfall.csv")
rain = pd.read_csv("Data/crop_production/Rainfall Predicted.csv")
data = df.dropna()
print(data.shape)
dis=d
state=list(data[data["District_Name"]==dis.upper()]["State_Name"][:1])[0]
season=se
area_in=ar
s=list(data["Season"].unique())
for x in s:
    if season.title() in x:
        sin=s.index(x)
data_cu=data[data["District_Name"]==dis.upper()][data["Season"]==s[sin]]
data1 = data_cu.drop(["State_Name","Crop_Year"],axis=1)
data_dum = pd.get_dummies(data1)
x = data_dum.drop("Production",axis=1)
y = data_dum[["Production"]]

model = RandomForestRegressor()
model.fit(x,y.values.ravel())

ch=pd.DataFrame()
for crop in list(data_cu["Crop"].unique()):
    t=(x[x["Crop_{}".format(crop)]==1])[:1]
```

```
        ch=pd.concat([ch,t])
    ch["Area"]=area_in
    ch["Rainfall"]=list(rain[rain["State_Name"]==state]["Rainfall"])[0]
    predict=model.predict(ch)
    crname=data.loc[ch.index]["Crop"]
    crdata= {'Crop': list(crname),
            'Production': list(predict)}
    crpro = pd.DataFrame(crdata)
    crpro=crpro.sort_values(by=['Production'], ascending=False)
    print(crpro)
    fig=plt.figure()
    ax = fig.add_axes([0,0,1,1])
    tstr='Predicted Production in        District: '+dis.title()+'      Season: '+season.title()
    ax.set_title(tstr,fontsize=15)
    ax.set_ylabel('Production in Tones', fontsize=14)
    ax.set_xlabel('Crop', fontsize=13)
    ax.bar(list(crpro["Crop"])[:5], list(crpro["Production"])[:5])
    plt.savefig('plot.png', dpi=400, bbox_inches='tight')
    im = Image.open('plot.png').show()
#classifier = load_model('Trained_model.h5')
#classifier._make_predict_function()
crop_recommendation_model_path = 'Crop_Recommendation.pkl'
crop_recommendation_model    =    pickle.load(open(crop_recommendation_model_path,
'rb'))
app = Flask(_name_)
```

```
@ app.route('/fertilizer-predict', methods=['POST'])
def fertilizer_recommend():

    crop_name = str(request.form['cropname'])
    N_filled = int(request.form['nitrogen'])
    P_filled = int(request.form['phosphorous'])
    K_filled = int(request.form['potassium'])

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

    N_desired = df[df['Crop'] == crop_name]['N'].iloc[0]
    P_desired = df[df['Crop'] == crop_name]['P'].iloc[0]
    K_desired = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = N_desired- N_filled
    p = P_desired - P_filled
    k = K_desired - K_filled

    if n < 0:
        key1 = "NHigh"
    elif n > 0:
        key1 = "Nlow"
    else:
        key1 = "NNo"
```

```
if p < 0:
    key2 = "PHigh"
elif p > 0:
    key2 = "Plow"
else:
    key2 = "PNo"

if k < 0:
    key3 = "KHigh"
elif k > 0:
    key3 = "Klow"
else:
    key3 = "KNo"

abs_n = abs(n)
abs_p = abs(p)
abs_k = abs(k)

response1 = Markup(str(fertilizer_dict[key1]))
response2 = Markup(str(fertilizer_dict[key2]))
response3 = Markup(str(fertilizer_dict[key3]))
return render_template('Fertilizer-Result.html', recommendation1=response1,
recommendation2=response2, recommendation3=response3,
```

```
   diff_n = abs_n, diff_p = abs_p, diff_k = abs_k)
@app.route("/")
@app.route("/index.html")
def index():
    return render_template("index.html")
@app.route("/CropRecommendation.html")
def crop():
    return render_template("CropRecommendation.html")
@app.route("/FertilizerRecommendation.html")
def fertilizer():
    return render_template("FertilizerRecommendation.html")
@app.route("/CropProductionPrediction.html")
def pesticide():
    return render_template("CropProductionPrediction.html")
@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        d = (request.form['district'])
        se = (request.form['season'])
        ar = (request.form['area'])
        finding(d,se,ar)
        #return render_template('result_prediction.html')
@ app.route('/crop_prediction', methods=['POST'])
def crop_prediction():
```

```
    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['potassium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])
        temperature = float(request.form['temperature'])
        humidity = float(request.form['humidity'])
        data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
        my_prediction = crop_recommendation_model.predict(data)
        final_prediction = my_prediction[0]
        return render_template('crop-result.html', prediction=final_prediction,
pred='img/crop/'+final_prediction+'.jpg')
if _name_ == '_main_':
    app.run(debug=True)
```

5.4 OUTPUT SCREENS

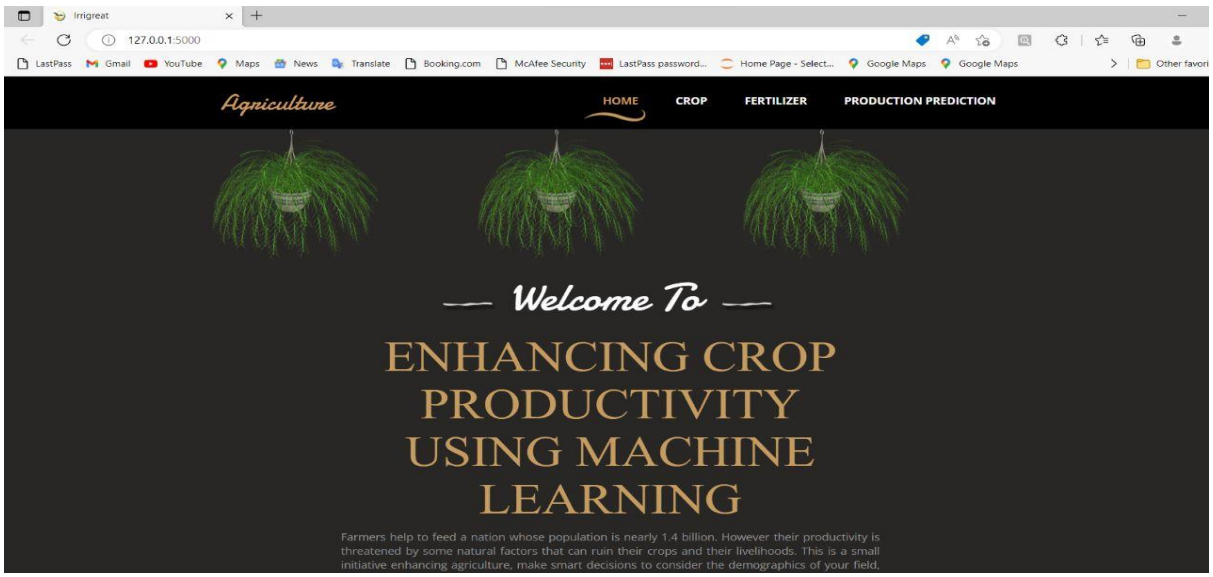

Figure5.4.1: *Home page*

Firstly, the user goes to home page where it contains three modules crop recommendation, fertilizer recommendation and crop production prediction.

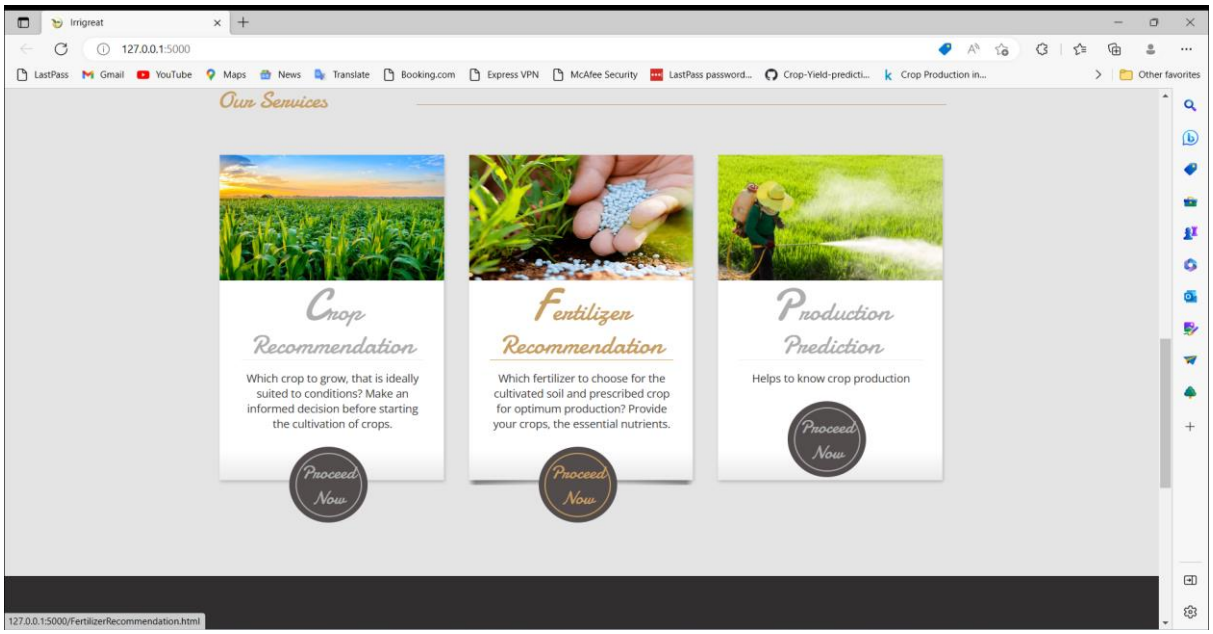

Figure5.4.2*: Recommendation Modules*

User selects crop recommendation module

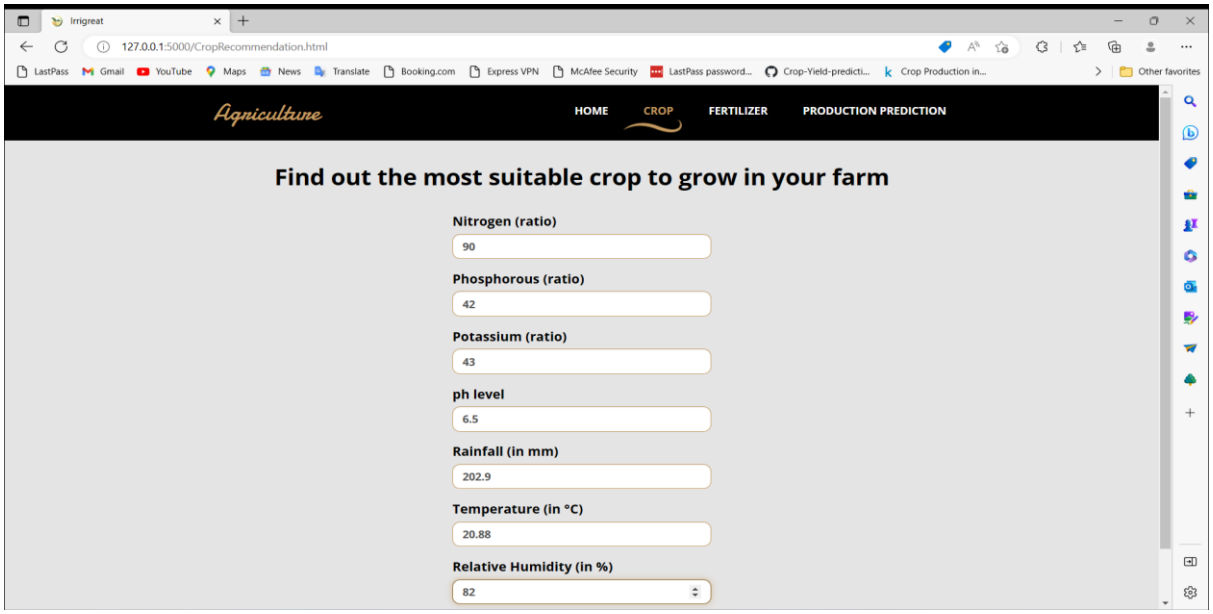

***Figure5.4.3**: Crop Recommendation*

User need to enter all the parameters namely N, P, K, pH, rainfall, temperature and humidity.

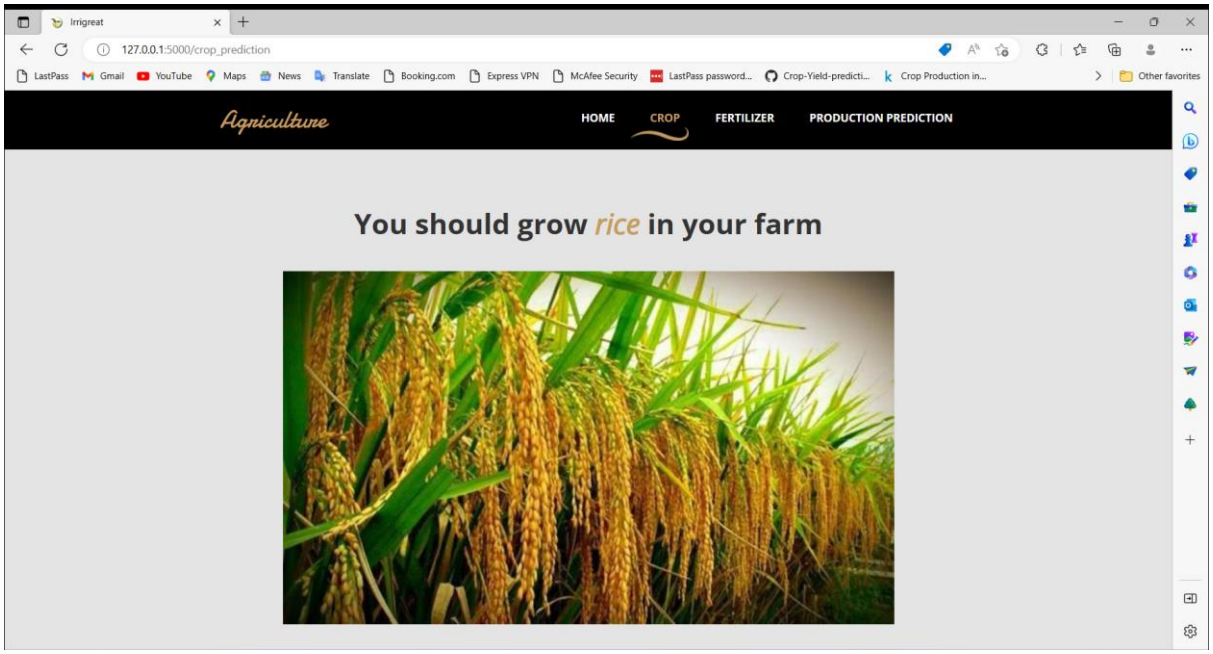

***Figure5.4.4**: Output of Crop Recommendation*

It displays the result to which crop should grow in your farm

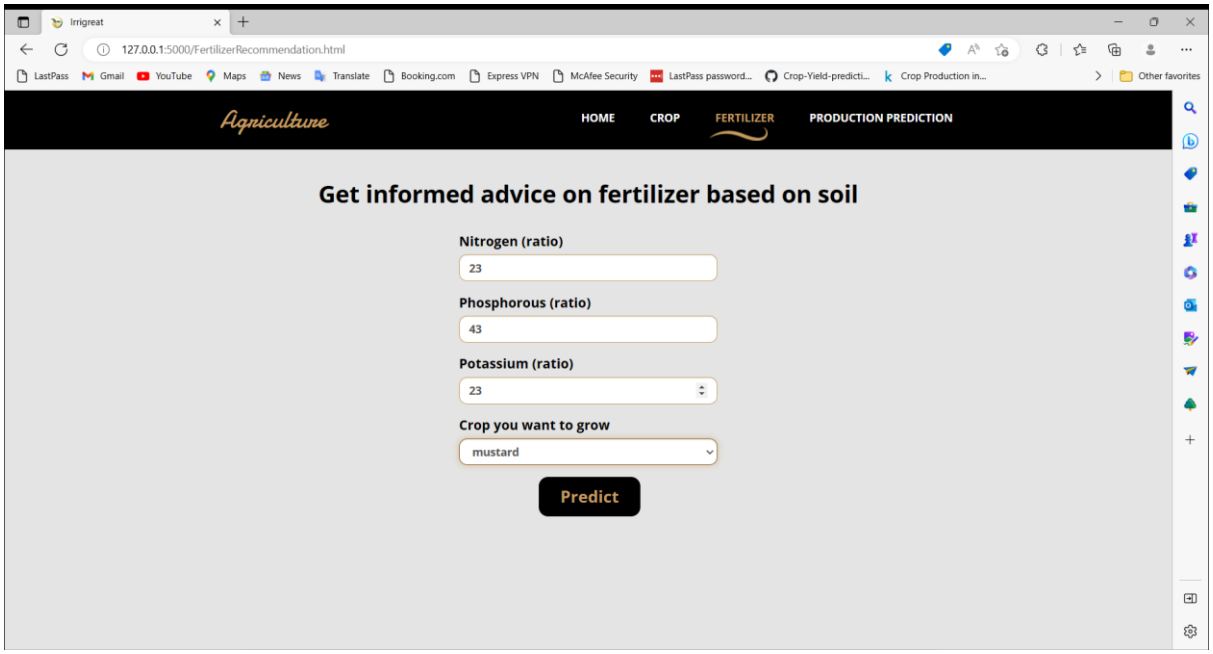

***Figure5.4.5**: fertilizer Recommendation*

User need to enter all the parameters related to fertilizer recommendation

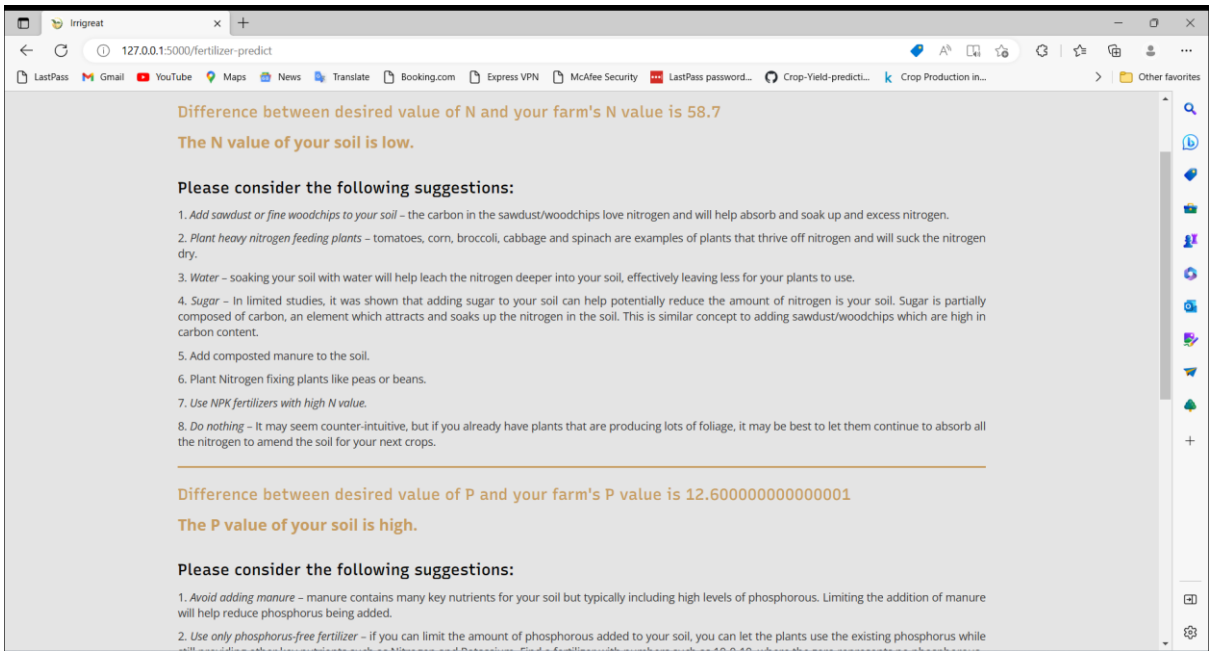

***Figure5.4.6**: Output of fertilizer Recommendation*

It displays the result to which fertilizer to use in your farm

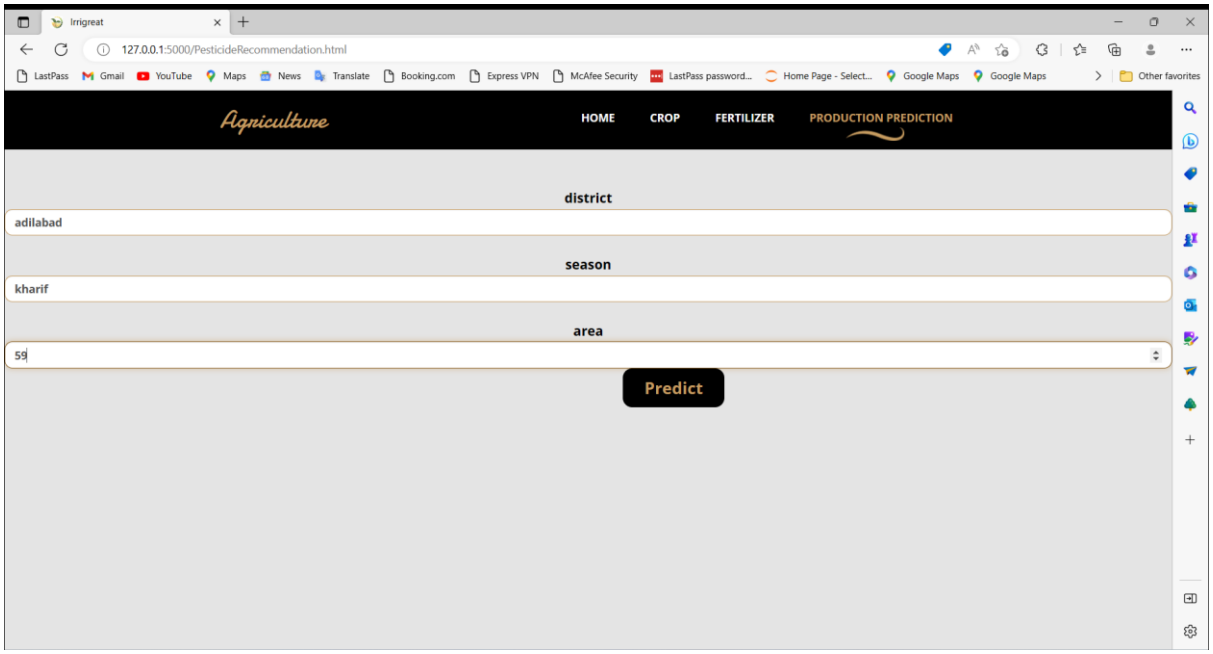

***Figure5.4.7**: Prediction of Crop Production*

User need to enter all the parameters related to crop production prediction

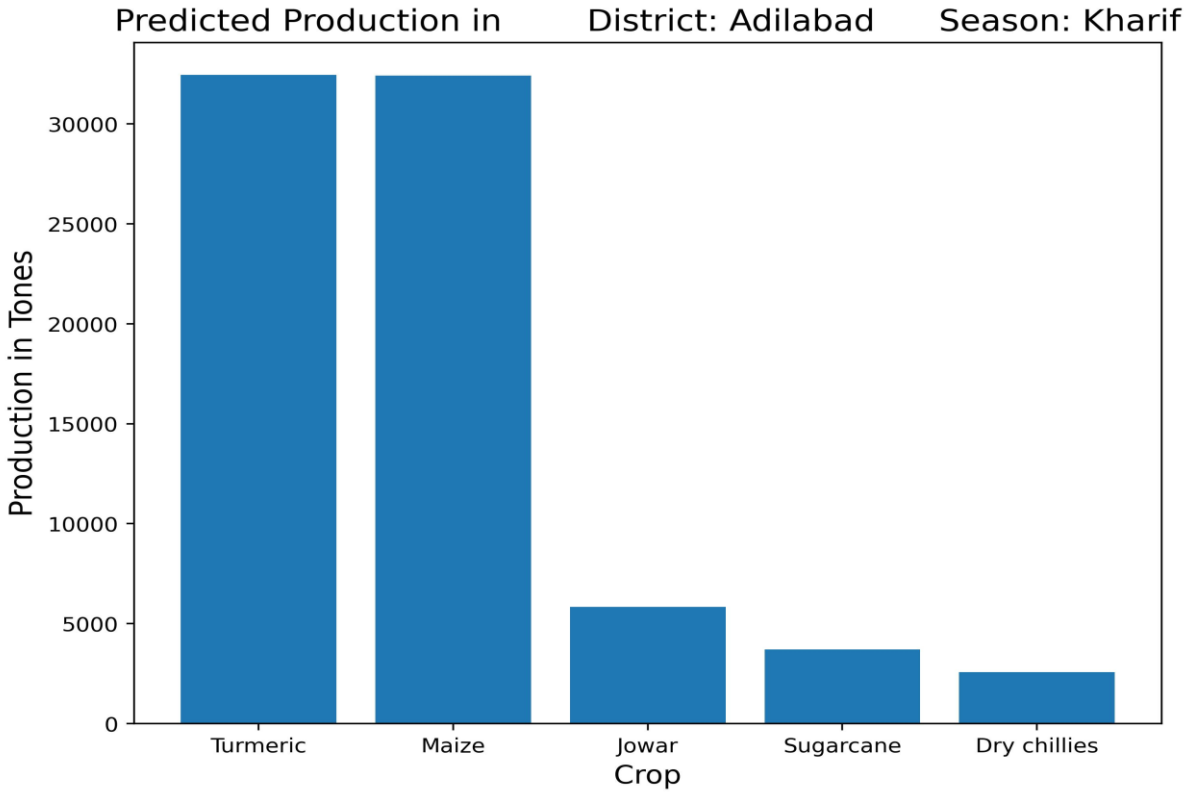

***Figure5.4.8:** Output of Prediction of Crop Production*

It displays the crop production to the user

6. TESTING

6.1 INTRODUCTION

Software Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a white box and Black Box Testing.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application.

Software Testing can be done in two ways:

1. **Verification:** It refers to the set of tasks that ensure that software correctly implements a specific function.
2. **Validation:** It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Importance of Software Testing:

The importance of software testing is imperative. Software Testing is important because of the following reasons:

1. Software Testing points out the defects and errors that were made during the development phases. It looks for any mistake made by the programmer during the implementation phase of the software.
2. It ensures that the customer finds the organization reliable and their satisfaction in the application is maintained. Sometimes contracts include monetary penalties with respect to the timeline and quality of the product and software testing prevent monetary losses.
3. It also ensures the Quality of the product. Quality product delivered to the customers helps in gaining their confidence. It makes sure that the software

application requires lower maintenance cost and results in more accurate, consistent and reliable results.

6.2 DESIGN OF TEST CASES AND SCENARIOS

6.2.1 TEST CASE DESIGN:

The design of tests for software and other engineering products can be as challenging as the initial design of the product. Test case methods provide the developer with a systematic approach to testing.

Any Engineered product can be tested in either of the two ways:

1. Knowing the specified function that a product has been designed to perform, tests can be conducted. These tests demonstrate whether each function is full operational and at the same time searches for errors in each function.
2. Knowing the internal workings of a product, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components hence been adequately exercised.

Test case design methods are divided into two types:

1. White-box testing
2. Black-box testing

1. **White-Box Testing**

White –box testing, sometimes called glass-box testing is a test, case designed method that uses the control structure of the procedural design to derive test cases. Using white-box testing methods, the s/w engineer can derive test cases that guarantee that all independent paths within a module have been exercised at least once. Exercise all logical decisions on their true and false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structures to ensure their validity.

2. **Black-Box Testing**

Black-box testing, also called behavioural testing, focuses on the functional requirements of the s/w. Black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of a program. It is a complementary approach that is likely to uncover a different class of errors that white-box methods could not.

6.2.2 SCENARIOS

The following are the test cases in our project:

Test **Objective:** To test individual module of system.

***Table 6.2.2**: Test Cases*

Test case Id	Test case description	Expected output	Actual output	Status
1	User go to home page	User can access all fields	User accessed all fields	success
2	If N value<0	Should display a message - value must be greater than or equal to 0.	Display a message-value must be greater than or equal to 0	success
3	If N value>100	Should display a message - value must be less than or equal to 100	Display a message- value must be less than or equal to 100	success
4	If P value<0	Should display a message - value must be greater than or equal to 0	Display a message- value must be greater than or equal to 0	success
5	If P value>100	Should display a message - value must be less than or equal to 100	Display a message-value must be less than or equal to 100	success
6	If K value<0	Should display a message - value must be greater than or equal to 0	Display a message-value must be greater than or equal to 0	success
7	If K value>100	Should display a message - value must be less than or equal to 100	Display a message-value must be less than or equal to 100	success
8	If pH>14	Should display a message - value must be less than or equal to 14	Display a message-value must be less than or equal to 14	success
9	If pH<0	Should display a message - value must be greater than or equal to 0.	Display a message-value must be greater than or equal to 0	Success

10	If Rainfall < 0	Should display a message - value must be greater than or equal to 0	Display a message-value must be greater than or equal to 0	Success
11	If Temperature < 0	Should display a message - value must be greater than or equal to 0	Display a message-value must be greater than or equal to 0	success
12	If Temperature > 100	Should display a message - value must be less than or equal to 100	Display a message-value must be less than or equal to 100	success
13	If Humidity < 0	Should display a message - value must be greater than or equal to 0	Display a message-value must be greater than or equal to 0	success
14	If Humidity > 100	Should display a message - value must be less than or equal to 100	Display a message-value must be less than or equal to 100	success
15	If any field is empty	Should display a message – Please fill out this field.	Display a message-Please fill out this field	success

7. CONCLUSION

7.1 PROJECT CONCLUSION

We have implemented a system that mainly focuses on agricultural productivity based on machine learning algorithms for prediction and recommendation. This will benefit farmers to maximize productivity in agriculture, reduce soil degradation in cultivated fields and have informed advice on organic fertilizers/ other fertilizers and also know about the right crop by considering various attributes. This would provide a comprehensive prediction and hence benefit both farmers and the environment. The inbuilt recommender system allows a user exploration of the possible crops and their yield to take more educated decisions In this way, we hope for the advancement of our agricultural economy, increasing productivity, and contributing to the prosperity of the country.

7.2 FUTURE ENHANCEMENT

In future, we can collect all the required data by using GPS technology location of land and by taking access from rain forecasting by government, we can also predict crop by just giving the GPS location. In Crop Recommendation, values are manually entered by user of temperature, humidity, rainfall. Admin can also use some weather API to fetch the real time parameters by the city and state.

8. REFERENCES

8.1 PAPER REFERENCES

[1] Paul, M., Vishwakarma, S. K., & Verma, A. (2015). Analysis of Soil Behavior and Prediction of Crop Yield Using Data Mining Approach. 2015 International Conference on Computational Intelligence and Communication Networks (CICN).

[2] Gandhi, N., Armstrong, L. J., Petkar, O., & Tripathy, A. K. (2016). Rice crop yield prediction in India using support vector machines. 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE).

[3] Umamaheswari S, Sreeram S, Kritika N, Prasanth DJ, "BIoT: Blockchain-based IoT for Agriculture", 11th International Conference on Advanced Computing (ICoAC), 2019 Dec 18 (pp. 324-327). IEEE.

[4] S. Pavani, Augusta SophyBeulet P., "Heuristic Prediction of Crop Yield Using Machine Learning Technique", International Journal of Engineering and Advanced Technology (IJEAT), December 2019, pp(135-138).

8.2 TEXT BOOKS

- Software Engineering, A practitioner's Approach-Roger S. Pressman,6th edition, Mc Graw Hill International Edition.
- The unified modelling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
- Software Testing techniques – Baris Beizer, Dreamtech, second edition

8.3 WEBSITES

- https://journalppw.com/index.php/jpsp/article/download/5496/3634/6321
- https://www.sciencedirect.com/science/article/pii/S0168169920302301
- https://www.researchgate.net/publication/353980529_Enhancement_Of_Agriculture_Based_Crop_Yield_Prediction_Using_R_Tool_And_Machine_Learning