**A Project Report on**

**TITLE**

**Industrial Internship Project report submitted in partial fulfilment of the Requirements for the award of the degree in**

**BACHELOR OF TECHNOLOGY**

**IN**

## COMPUTER SCIENCE AND ENGINEERING

### BY

**Project Associates: (Batch -12)**

**Y.Gowthami 218X1A05H0**

**I.Sai Sri Ram 218X1A05J2**

**B.Umar 218X1A05I5**

**T.Sai Kishore reddy 218X1A05I0**

### Under the Esteemed Guidance of

**Mrs.CH. Sowjanya**

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)**

### ACCREDITED BY NBA & NAAC WITH ‘A’ GRADE (APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA) NH-5, CHOWDAVARAM, GUNTUR - 522019

**2020** - **2024**

## KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

### ACCREDITED BY NBA & NAAC WITH ‘A’ GRADE (APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA) NH-5, CHOWDAVARAM, GUNTUR - 522019

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the Industrial Internship Project work entitled **Fertilizer Recommendation System For Agriculture Using AI** being submitted by

**Y.Gowthami 218X1A05H0**

**I.Sai Sri Ram 218X1A05J2**

**B.Umar 218X1A05I5**

**T.Sai Kishore reddy 218X1A05I0**

in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Kallam Haranadhareddy Institute of Technology is a record of bonafide work carried out by them.

|  |  |
| --- | --- |
| **Internal Guide**  **Mrs.CH. Sowjanya**  Assistant Professor | **Head of the Department**  **Dr. V. RAJIV JETSON**  Professor & HOD |

**External Examiner**

# CERTIFICATE FROM INTERN ORGANIZATION

**DECLARATION**

### We Y.Gowthmi (218X1A05H0), I.Sai SriRam (218X1A05j2), B. Umar (218X1A05I5), T.Sai Kishore Reddy (218X1A05IO) hereby declare that the project report titled “Fertilizer Recommendation System For Agriculture Using AI” under the guidance of Mrs. CH. Sowjanya, M.Tech Assistant Professor is submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering.

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project have not been submitted to any other university for the award of any degree.

**Y.Gowthami 218X1A05H0**

**I.Sai Sri Ram 218X1A05J2**

**B.Umar 218X1A05I5**

**T.Sai Kishore reddy 218X1A05I0**

**ACKNOWLEDGEMENT**

We profoundly grateful to express our deep sense of gratitude and respect towards **Sri. KALLAM MOHAN REDDY**, **Chairman, KHIT**, **Guntur** for his precious support in the college**.**

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, GUNTUR for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY**, principal, KHIT, GUNTUR for his support during and until the completion of the project.

We are greatly indebted to **Dr. V. RAJIV JETSON, M.Tech, ph. D** Professor & Head of the department, Computer Science and Engineering KHIT, GUNTUR for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college during Industry Internship.

We are also thankful to our Project Coordinators **MR. G. MANTHRU NAIK** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Internal Guide

**Mrs.CH. Sowjanya**  **M.Tech**, Assistant Professor and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of great help towards the successful completion of our project.

**Y.Gowthami 218X1A05H0**

**I.Sai Sri Ram 218X1A05J2**

**B.Umar 218X1A05I5**

**T.Sai Kishore reddy 218X1A05I0**

# ABSTRACT

The project focuses on developing an AI-driven fertilizer recommendation system utilizing a comprehensive dataset sourced from Kaggle, which includes critical agronomic parameters such as Temperature, Humidity, Moisture, Soil Type, Crop Type, Nitrogen, Potassium, Phosphorous, and the corresponding Fertilizer Name. The process begins with thorough data preprocessing and exploratory data analysis (EDA) to ensure the quality and relevance of the input data, facilitating a more accurate model performance. Two machine learning algorithms, namely the Random Forest classifier and the Multi-Layer Perceptron (MLP) classifier, are employed to predict optimal fertilizer recommendations based on the analyzed features. The Random Forest classifier achieves an impressive accuracy of 100%, demonstrating its effectiveness in handling the complexities of the dataset, while the MLP classifier also performs admirably with a 95% accuracy rate. To enhance user accessibility and interaction, a Flask-based user interface (UI) is developed, enabling real-time fertilizer predictions based on user-inputted agronomic data. This project not only contributes to precision agriculture but also aims to promote sustainable farming practices by optimizing fertilizer usage, thereby reducing environmental impact and enhancing crop yield.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO** |
| Abstract | i |
| Table of Contents | ii |
| List of Figures | v |
| **Chapter 1 Introduction** | 1-3 |
| 1.1 Objective of the Project | 2 |
| 1.2 Problem Statement | 2 |
| 1.3 Motivation | 2 |
| 1.4 Scope | 3 |
| 1.5 Project Introduction | 3 |
| **Chapter 2 Literature Survey** | 4-6 |
| 2.1 Related Work | 5 |
| **Chapter 3 System Analysis** | 7-9 |
| 3.1 Existing System | 8 |
| 3.2 Disadvantages | 8 |
| 3.3 Proposed System | 9 |
| 3.4 Advantages | 9 |
| 3.5 work Flow of Proposed system | 10 |

|  |  |
| --- | --- |
| **Chapter 4 Requirement Analysis** | 11-14 |
| 4.1 Function and non-functional requirements | 12 |
| 4.2 Software Requirements | 13 |
| 4.3 Hardware Requirements | 14 |
| 4.4 Architecture | 14 |
| **Chapter 5 Methodology** | 15-19 |
| 5.1 Random Forest Classifier | 16 |
| 5.2 MLP Classifier | 18 |
| **Chapter 6 System Design** | 20-28 |
| 6.1 Introduction of Input design | 21 |
| 6.2 UML Diagrams | 22 |
| 6.3 Data Flow Diagram | 28 |
| **Chapter 7 Implementation and Results** | 29-41 |
| 7.1 Modules | 30 |
| 7.2 Coding | 32 |
| 7.3 Output Screens | 39 |
| **8. System Study and Testing** | 42-47 |
| 8.1 Feasibility study | 43 |
| 8.2 Types of testing | 44 |
| 8.3 Test Cases | 47 |
| **9. Result** | 49 |
| **10. Conclusion** | 51 |
| **11. Future Enhancement** | 53 |
| **12. References** | 55 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **NAME** | **PAGE NO** |
| 3.5.1 Project Flow | 10 |
| 4.4.1 Project Architecture | 14 |
| 6.2.1 Use case diagram | 22 |
| 6.2.2 Class diagram | 23 |
| 6.2.3 Sequence diagram | 24 |
| 6.2.4 Collaboration diagram | 25 |
| 6.2.5 Deployment diagram | 25 |
| 6.2.6 Activity diagram | 26 |
| 6.2.7 Component diagram | 27 |
| 6.2.8 ER diagram | 27 |
| 6.3.1 Context diagram | 28 |
| 7.3.1 Count of Fertilizers | 39 |
| 7.3.2 Distribution of Features | 39 |
| 7.3.3 Model Performance Comparison | 40 |
| 7.3.4 Home page | 40 |
| 7.3.5 Output Predictions | 41 |
| 7.3.6 Output Predictions | 41 |

# CHAPTER 1 INTRODUCTION

1. **INTRODUCTION**

## OBJECTIVE OF PROJECT:

The objective of this project is to develop an intelligent fertilizer recommendation system that leverages machine learning techniques to provide tailored fertilizer suggestions based on key agronomic parameters. By analyzing factors such as Temperature, Humidity, Moisture, Soil Type, Crop Type, Nitrogen, Potassium, and Phosphorous, the system aims to optimize fertilizer application, enhance crop yield, and promote sustainable agricultural practices. Through a user-friendly Flask-based interface, farmers and agricultural stakeholders will have easy access to precise fertilizer recommendations, ultimately contributing to more efficient farming and reduced environmental impact.

## PROBLEM STATEMENT:

The agricultural sector faces significant challenges in determining the appropriate fertilizer types and application rates, which can lead to inefficient fertilizer use, reduced crop yield, and negative environmental impacts. Farmers often lack access to timely and precise information regarding fertilizer recommendations tailored to specific soil and crop conditions. This project aims to address this issue by developing an AI-based fertilizer recommendation system that utilizes machine learning algorithms to analyze key agronomic factors, providing accurate and personalized fertilizer suggestions.

## MOTIVATION:

* + - **Food Security**: With the global population steadily increasing, there is an urgent need to enhance agricultural productivity to ensure food security for future generations.
    - **Sustainable Practices**: Promoting efficient fertilizer usage can significantly reduce environmental degradation, helping to maintain soil health and minimize water pollution.
    - **Cost Efficiency**: Providing accurate fertilizer recommendations can help farmers save on input costs by minimizing wastage and optimizing fertilizer application rates.
    - **Empowering Farmers**: By offering accessible and practical solutions, the project aims to empower farmers with the knowledge and tools needed to improve their crop yields and livelihoods.

## SCOPE:

The scope of this project encompasses the development of an AI-based fertilizer recommendation system that integrates various machine learning algorithms to analyze a diverse dataset of agronomic parameters, including Temperature, Humidity, Moisture, Soil Type, Crop Type, Nitrogen, Potassium, and Phosphorous. It involves comprehensive data preprocessing and exploratory data analysis (EDA) to ensure high-quality input for model training. The project will focus on implementing and evaluating multiple classifiers, specifically the Random Forest and Multi-Layer Perceptron algorithms, to identify the most effective model for predicting optimal fertilizer recommendations.

## PROJECT INTRODUCTION:

Agriculture is a critical sector that sustains the livelihoods of billions of people globally, yet it faces numerous challenges, including the need for increased productivity to meet the demands of a growing population, which is projected to reach 9.7 billion by 2050. According to the Food and Agriculture Organization (FAO), food production must increase by 70% to meet this demand, necessitating the optimal use of agricultural inputs such as fertilizers. However, improper fertilizer application can lead to adverse effects, including soil degradation, water contamination, and increased greenhouse gas emissions, emphasizing the urgent need for precise fertilizer management.

To address these challenges, this project focuses on developing an AI-driven fertilizer recommendation system utilizing a comprehensive dataset from Kaggle, encompassing critical factors such as Temperature, Humidity, Moisture, Soil Type, Crop Type, Nitrogen, Potassium, and Phosphorous. By employing advanced machine learning algorithms, including Random Forest and Multi-Layer Perceptron classifiers, the project aims to provide accurate and personalized fertilizer suggestions based on real-time agronomic data. Furthermore, a user-friendly Flask-based interface will be developed to enhance accessibility for farmers and agricultural stakeholders, facilitating informed decision-making and promoting sustainable agricultural practices that can ultimately contribute to increased crop yields and improved food security.

.

# CHAPTER 2 LITERATURE SURVEY

1. **LITERATURE SURVEY**

## RELATED WORK:

### " A Machine Learning Approach for Fertilizer Recommendation" by Kumar et al.

This paper investigates the use of machine learning algorithms for recommending fertilizers based on soil and crop parameters. Author A applies various algorithms, including Decision Trees and Support Vector Machines, to predict optimal fertilizer types and quantities. The study emphasizes the importance of utilizing local soil and crop data to enhance recommendation accuracy and relevance.

### Summary:

The research findings indicate that machine learning models can effectively recommend fertilizers, achieving high accuracy rates in predictions. The authors conclude that integrating machine learning into agricultural practices can lead to improved crop yields and more sustainable fertilizer usage.

### " Predicting Fertilizer Requirement Using Data Mining Techniques" by Sharma et al.

This paper presents a comprehensive analysis of data mining techniques to predict fertilizer requirements based on environmental factors. Author A employs multiple regression analysis and classification algorithms to assess the relationship between soil characteristics, weather conditions, and fertilizer needs.

### Summary:

The results show that data mining techniques can successfully predict fertilizer requirements, with regression models yielding the most accurate predictions. The authors highlight the potential for these techniques to optimize fertilizer application and promote sustainable farming practices.

### " Application of Artificial Intelligence in Precision Agriculture: A Review" by Patel et al.

This review article examines the role of artificial intelligence (AI) in precision agriculture, focusing on various applications, including fertilizer recommendation systems. Author A discusses the integration of AI with remote sensing and IoT technologies to enhance agricultural decision-making processes.

### Summary:

The authors conclude that AI technologies can significantly improve agricultural productivity by providing precise recommendations tailored to specific environmental conditions. The paper underscores the importance of developing robust AI models that consider diverse agronomic factors for effective fertilizer management.

### " Deep Learning for Fertilizer Recommendation: A Case Study" by Gupta et al.

This case study explores the application of deep learning techniques for fertilizer recommendation, utilizing a neural network model trained on extensive agricultural datasets. Author A investigates the model's performance in predicting fertilizer requirements based on various input parameters such as soil nutrients and crop types.

### Summary:

The findings indicate that deep learning models outperform traditional methods in accuracy and adaptability for fertilizer recommendations. The authors advocate for the continued exploration of deep learning in agriculture, emphasizing its potential to revolutionize fertilizer management practices and enhance crop yields.

# CHAPTER 3 SYSTEM ANALYSIS

1. **SYSTEM ANALYSIS**

## EXISTING METHOD

Existing methods for fertilizer recommendation primarily rely on traditional agronomic practices and empirical knowledge, which often involve subjective assessments of soil and crop conditions. Common approaches include soil testing and expert consultations to determine the appropriate type and quantity of fertilizer to apply. While these methods can provide some level of guidance, they lack the precision and adaptability needed for optimizing fertilizer use across varying environmental conditions. Additionally, some studies have explored the application of regression models and basic machine learning techniques to predict fertilizer needs, but these approaches often fall short in accuracy and may not effectively account for the complex interactions between soil properties, crop types, and climatic factors. As a result, there is a growing need for advanced AI-driven solutions that can offer more accurate and tailored fertilizer recommendations based on comprehensive data analysis.

## DISADVANTAGES:

* + - **Limited Precision**: Traditional methods often rely on general recommendations based on broad soil and crop categories, leading to imprecise fertilizer applications that may not meet the specific needs of individual fields.
    - **Time-Consuming**: Soil testing and expert consultations can be time-intensive processes, delaying timely fertilizer application and potentially impacting crop yield.
    - **Subjectivity**: Recommendations based on agronomic expertise may vary between consultants, introducing a level of subjectivity and inconsistency in the advice provided to farmers.
    - **Inadequate Data Utilization**: Existing methods typically do not leverage the wealth of available data from modern agricultural practices, such as weather patterns, soil moisture levels, and crop growth stages, resulting in missed opportunities for optimizing fertilizer use.

## PROPOSED METHOD:

The proposed method involves developing an AI-driven fertilizer recommendation system that utilizes advanced machine learning algorithms to analyze a comprehensive dataset encompassing key agronomic parameters such as Temperature, Humidity, Moisture, Soil Type, Crop Type, Nitrogen, Potassium, and Phosphorous. By employing models like Random Forest and Multi-Layer Perceptron (MLP), the system aims to provide accurate and tailored fertilizer recommendations based on real-time inputs, effectively addressing the limitations of traditional methods. The integration of data preprocessing and exploratory data analysis (EDA) will ensure high-quality input for model training, while a user-friendly Flask-based interface will facilitate easy access for farmers and agricultural stakeholders. This innovative approach not only enhances the precision of fertilizer application but also promotes sustainable agricultural practices by optimizing resource use and minimizing environmental impact.

## ADVANTAGES:

* + - **Increased Accuracy**: The use of advanced machine learning algorithms enables the system to provide precise fertilizer recommendations tailored to specific soil and crop conditions, improving overall application efficiency and crop yields.
    - **Data-Driven Insights**: By leveraging comprehensive datasets and real-time inputs, the proposed method can analyze complex interactions between various agronomic factors, leading to more informed decision-making and optimized fertilizer use.
    - **User-Friendly Interface**: The development of a Flask-based interface makes the system accessible to farmers and agricultural stakeholders, allowing them to easily input data and receive personalized recommendations, ultimately enhancing user engagement and adoption of sustainable practices.

## PROJECT FLOW

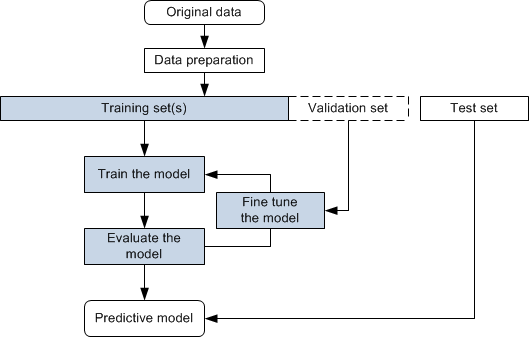
****

Fig 3.5.1 Project Flow

# CHAPTER 4 REQUIREMENTS ANALYSIS

1. **REQUIREMENTS ANALYSIS**

## FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types:

* + - Functional
    - Non-Functional Requirements

**Functional Requirements:** These are the requirements that end user specifically demands as basic facilities that a system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

1. **Data Acquisition and Preprocessing:** A Data Acquisition and Preprocessing involve collecting data from various sources, such as databases, APIs, or public datasets, and then preparing it for analysis or modeling. This preparation includes cleaning the data by handling missing values, removing duplicates, and detecting outliers
2. **Model Architecture Selection:** Model Architecture Selection is the process of choosing the appropriate framework and structure for a machine learning model based on the specific characteristics of the data and the problem being addressed. This involves considering various architectures, such as linear models, decision trees, or deep learning frameworks.
3. **Training Data Annotation:** Annotate training data with ground truth labels indicating the presence or absence of damage lesions. Ensure accuracy and consistency in annotation to facilicate model training.
4. **Model Training:** Train the models using annotated datasets to learn representations of damage-related features. Optimize hyper-parameters and model architectures to improve performance metrics such as accuracy, sensitivity and specificity.

**Non-Functional Requirements:** These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these

factors are implemented varies from one project to other.

1. **Scalability:** Horizontal scalability design ensures the system to scale horizontally across multiple nodes or servers to handle increased workload and data volume. Vertical scalability ensures that the system can scale vertically by upgrading hardware resources to meet growing
2. **Reliability:** The system should be 90% reliable. Since it may need some maintenance or preparation for some particular day, the system does not need to be reliable every time. So, 80% reliability is enough.
3. **Availability:** It is available to all Insurance companies.
4. **Cost Efficiency:** Design the system to minimize costs associated with hardware, software, maintenance, training and return on investment is to evaluate the system’s ROI by considering its effectiveness, cost savings and other benefits compared to traditional damage detection methods.

## SOFTWARE REQUIREMENS

Operating System : Windows 7/8/10

Server side Script : HTML, CSS & JS

Programming Language : Python

Libraries : Flask, Pandas, Tensorflow, Keras, Sklearn, Numpy

IDE/Workbench : VSCode

Technology : Python 3.11.4

## HARDWARE REQUIREMENTS

Processor - I3/Intel Processor

RAM - 8GB (min)

Hard Disk - 128 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - Any

## ARCHITECTURE:

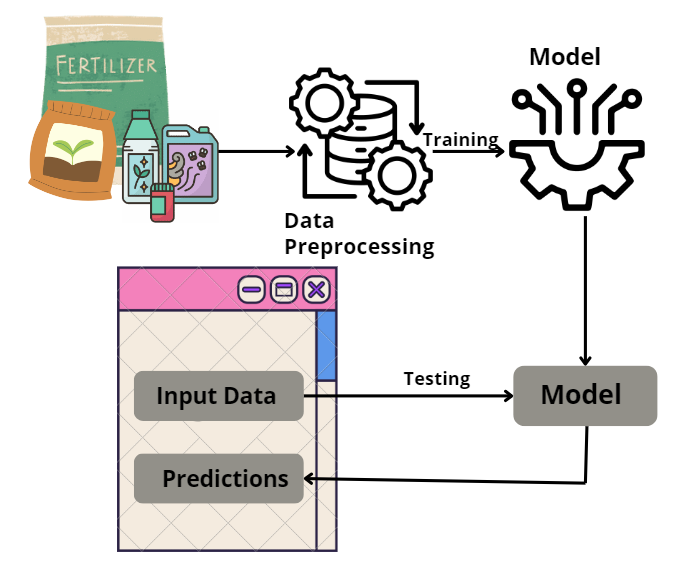


Fig 4.4.1 Project Architecture

# CHAPTER 5 METHODOLOGY

1. **METHODOLOGY**

## Random Forest Classifier

Random Forest is an ensemble learning method primarily used for classification and regression tasks in machine learning. Developed by Leo Breiman and Adele Cutler in the early 2000s, it operates by constructing a multitude of decision trees during training and outputting the mode of the classes for classification or the mean prediction for regression. This technique leverages the principles of bagging (Bootstrap Aggregating) to improve predictive accuracy and control overfitting, making it particularly effective for complex datasets with high dimensionality. The versatility and robustness of Random Forest have made it a popular choice in various applications, including finance, healthcare, and agriculture, where accurate predictions are crucial.

The core mechanism of Random Forest involves creating multiple decision trees from random subsets of the training data. Each tree is built using a different bootstrap sample, which is a random sample of the data with replacement. As a result, some observations may appear multiple times while others may not be included at all. Additionally, when splitting nodes, Random Forest considers a random subset of features rather than all features, promoting diversity among the trees and enhancing the model's generalization capabilities. The final prediction is made by aggregating the outputs of all the trees—using majority voting for classification tasks or averaging for regression tasks. This ensemble approach helps reduce variance and improve overall performance compared to individual decision trees.

One of the key advantages of Random Forest is its ability to handle large datasets with higher dimensionality without overfitting. The randomness introduced in both data sampling and feature selection helps the model generalize better, making it resilient to noise and outliers in the dataset. Additionally, Random Forest provides inherent measures of feature importance, allowing practitioners to identify the most influential variables in their predictive models. This capability is particularly valuable in fields like agriculture, where understanding which factors most affect outcomes can inform better management practices. Furthermore, the method is relatively easy to implement and tune, requiring fewer hyperparameter adjustments compared to more complex algorithms.

Random Forest has found extensive applications across various domains due to its robustness and versatility. In agriculture, it is frequently used for tasks such as crop yield prediction, soil classification, and fertilizer recommendation, as it can effectively analyze complex interactions among multiple environmental variables. In finance, the algorithm helps in credit scoring and fraud detection by assessing numerous risk factors simultaneously. Healthcare applications include disease prediction and diagnosis, where Random Forest can analyze patient data to identify significant health risks. Its flexibility allows it to be used in both structured and unstructured data scenarios, making it a valuable tool in the machine learning toolkit.

Despite its many advantages, Random Forest is not without limitations. One notable drawback is its relatively low interpretability compared to simpler models like linear regression. While feature importance metrics provide some insights, understanding the relationships captured by multiple decision trees can be challenging, which may hinder transparency in critical applications like healthcare. Additionally, Random Forest can be computationally intensive, especially with large datasets and a high number of trees, potentially leading to longer training times and increased memory usage. Lastly, while it generally performs well on many tasks, it may not always outperform specialized algorithms tailored for specific types of data or problems, such as deep learning methods for image or text data.

In conclusion, Random Forest is a powerful and flexible machine learning algorithm that excels in a wide range of applications due to its robustness, accuracy, and ability to handle complex datasets. Its ensemble approach, which combines the predictions of multiple decision trees, helps mitigate overfitting while providing reliable predictions. While there are limitations in interpretability and computational efficiency, its advantages often outweigh these concerns, making it a popular choice among data scientists and practitioners. As machine learning continues to evolve, Random Forest remains a foundational technique, particularly in fields that require effective analysis of intricate relationships among variables, such as agriculture and environmental science.

## Multi-Layer Perceptron (MLP) classifier

The Multi-Layer Perceptron (MLP) classifier is a type of feedforward artificial neural network (ANN) that consists of multiple layers of nodes or neurons, each connected to the neurons in the adjacent layers. Unlike traditional linear models, the MLP can model complex, non-linear relationships between input features and target labels by introducing hidden layers between the input and output layers. Each neuron in an MLP applies a weighted sum of its inputs, passes the result through an activation function, and outputs a value that becomes an input for the next layer. This structure enables the MLP to perform tasks such as classification, regression, and pattern recognition effectively. In the context of fertilizer recommendation systems, an MLP classifier is particularly useful because it can capture the intricate dependencies between multiple agronomic factors such as Temperature, Humidity, Soil Type, and Nutrient levels and predict the optimal fertilizer for a given crop. One of the key advantages of MLP classifiers is their ability to learn from large datasets and improve their performance over time.

During the training process, the MLP uses a method called backpropagation to update the weights of the connections between neurons. Backpropagation works by calculating the error at the output layer (i.e., the difference between the predicted output and the actual label), and then propagating this error backward through the network, adjusting the weights to minimize the error. This iterative process, typically combined with optimization algorithms like gradient descent, allows the MLP to learn complex patterns in the data. In the fertilizer recommendation project, this learning process enables the MLP to improve its predictions over time, providing more accurate fertilizer recommendations as more data becomes available. The MLP classifier also incorporates non-linearity through the use of activation functions, which are applied to the weighted sums calculated by the neurons. Common activation functions include the sigmoid function, hyperbolic tangent (tanh), and the rectified linear unit (ReLU). The introduction of non-linearity is essential for capturing complex relationships in the data that linear models cannot handle. For instance, in the fertilizer recommendation task, factors such as nutrient levels and soil types interact in non-linear ways to influence crop growth and fertilizer needs.

The MLP's use of non-linear activation functions allows it to model these interactions more effectively, leading to better prediction performance compared to simpler models. Despite its strengths, the MLP classifier also comes with challenges, particularly in terms of training and computational complexity. Since MLPs have multiple layers and many parameters (weights and biases), they require significant computational resources, especially for large datasets. Training an MLP can be time-consuming, as the network needs to process the data multiple times (epochs) to converge on an optimal solution. Additionally, MLPs are prone to overfitting, especially when the network is too complex or the training data is limited. Overfitting occurs when the model learns to memorize the training data, including noise, rather than generalizing to unseen data. To mitigate overfitting, techniques such as dropout (randomly disabling neurons during training) and early stopping (halting training once performance on a validation set starts to degrade) are commonly employed. In the context of the fertilizer recommendation system, careful tuning of the MLP’s hyperparameters such as the number of hidden layers, the number of neurons per layer, and the learning rate is necessary to strike a balance between accuracy and computational efficiency.

One of the most important considerations when using MLP classifiers is the selection of input features. The performance of an MLP is highly dependent on the quality and relevance of the input data. For the fertilizer recommendation system, input features might include soil characteristics, crop type, weather conditions, and nutrient levels. Feature selection techniques, such as Principal Component Analysis (PCA) or Recursive Feature Elimination (RFE), can help reduce the dimensionality of the data and improve the efficiency of the MLP by eliminating irrelevant or redundant features. By focusing on the most important features, the MLP can learn more effectively and provide more accurate predictions without being overwhelmed by unnecessary data. In the fertilizer project, the choice of input features is critical for ensuring that the MLP can accurately capture the factors that influence fertilizer requirements.

# CHAPTER 6 SYSTEM DESIGN

1. **SYSTEM DESIGN**

## INTRODUCTION OF INPUT DESIGN:

The Input Design component focuses on the methods and processes for preparing and structuring input data for the multi perspective Fertilizer Predictions. This includes preprocessing, extracting relevant features, and formatting the input for effective processing by Machine Learning Algorithms.

## Objectives for Input Design:

* Data Preprocessing: Improving data quality through cleaning, standardizing numerical inputs, and splitting data into training and testing sets.
* Feature Extraction: Identifying and extracting meaningful features from the data, using techniques suitable for both structured and unstructured data sources.
* Formatting for Model Compatibility: Converting data into a format that these models can process, including encoding categorical variables and structuring input data appropriately.

## Output Design:

Output Design refers to the process of defining and structuring the results generated by a model or system to ensure they are clear, relevant, and actionable for end-users. This involves determining the format, content, and presentation of the output, which may include visualizations, reports, dashboards, or user interfaces that effectively convey the insights derived from the data. A well-designed output enhances user experience, facilitates decision-making, and ensures that the results align with the intended goals of the project or application. Additionally, incorporating contextual relevance, feedback mechanisms, and performance metrics allows users to understand and apply the outputs effectively. Overall, well-designed outputs empower users to make informed decisions based on the insights generated, bridging the gap between complex analysis and practical application.

## UML DIAGRAMS:

### USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

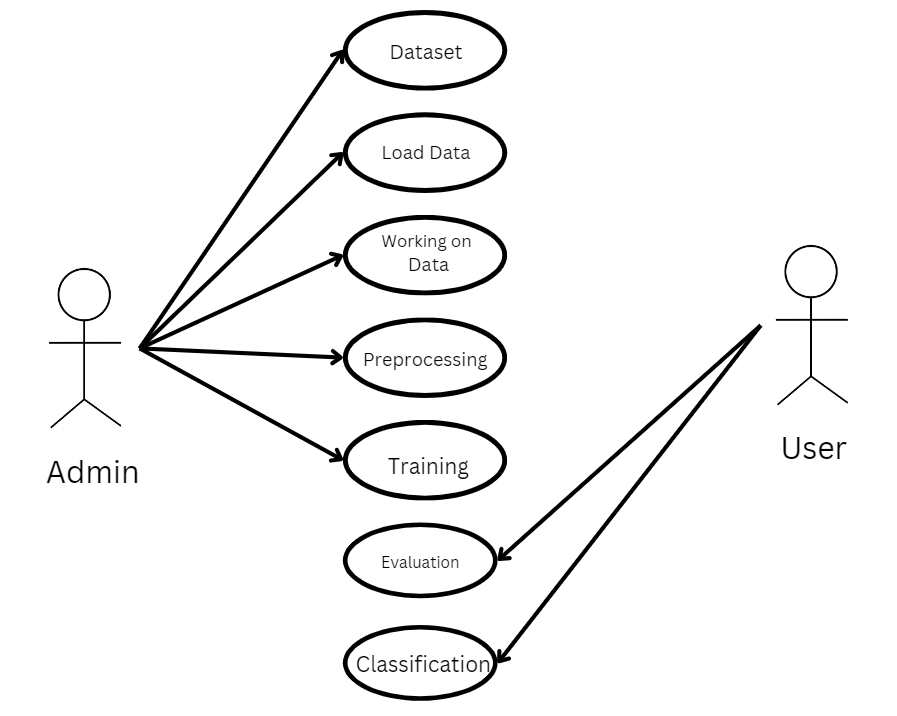


Fig 6.2.1 Use case diagram

### CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

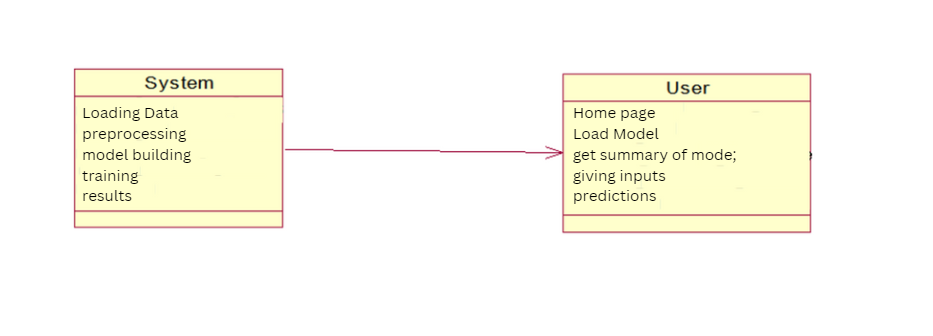


Fig 6.2.2 Class diagram

### SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

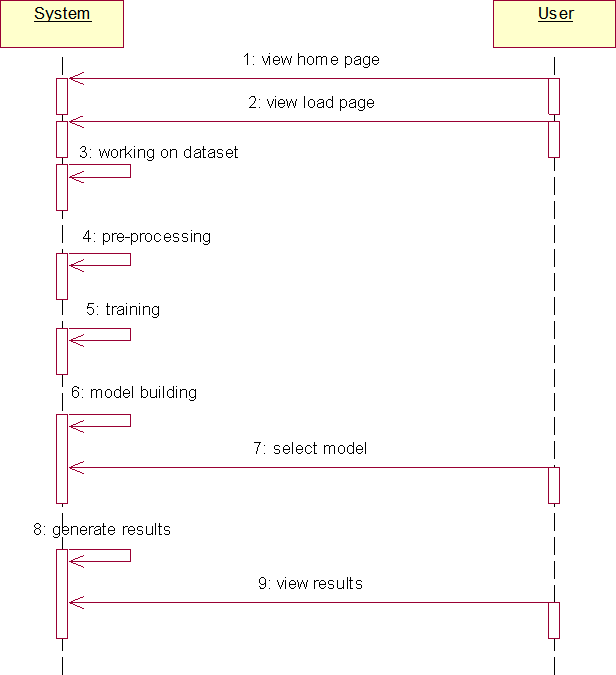


Fig 6.2.3 Sequence diagram

### COLLABRATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

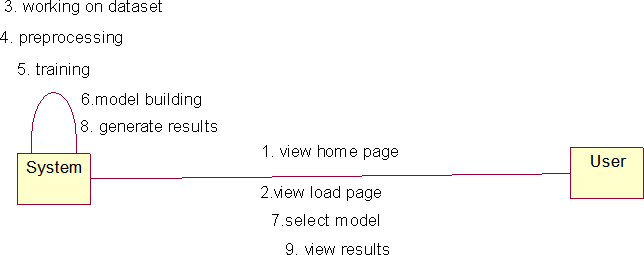


Fig 6.2.4 Collaboration diagram

### DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



Fig 6.2.5 Deployment diagram

### ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

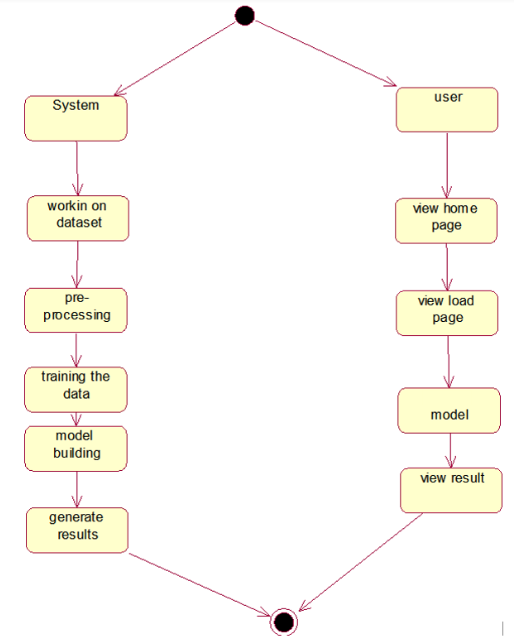


Fig 6.2.6 Activity diagram

### COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by

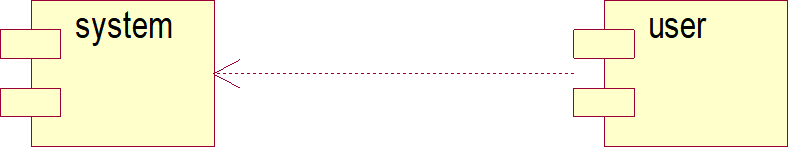


Fig 6.2.7 Component diagram

### ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.

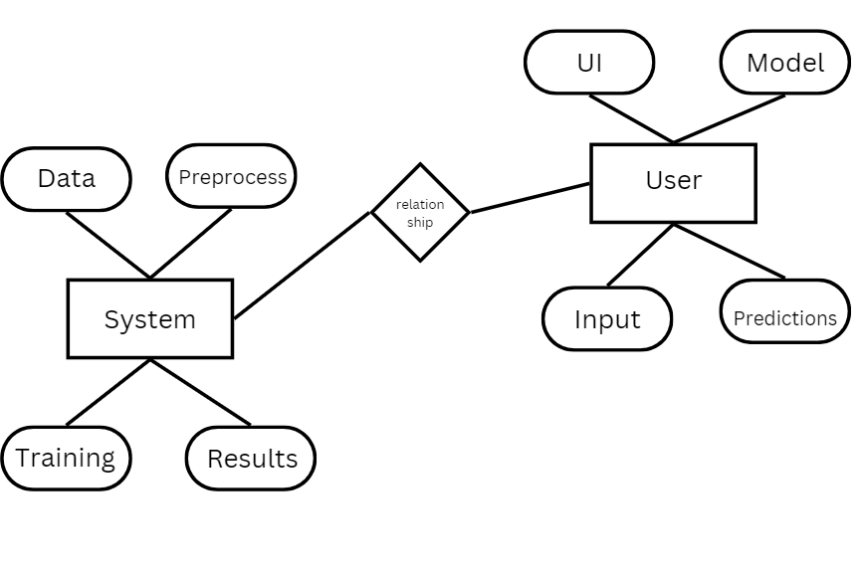


Fig 6.2.8 ER diagram

## DFD DIAGRAM

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

# Context Diagram:

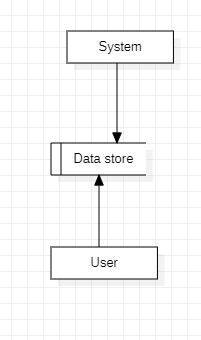


Fig 6.3.1 Context diagram

# CHAPTER 7 IMPLEMENTATION AND RESULTS

1. **IMPLEMENTATION AND RESULTS**

## MODULES

1. **System:**

### Preprocessing:

Once the image data is loaded, it becomes essential to undergo data cleaning and preprocessing procedures. This involves tasks like handling potential image artifacts, addressing missing or corrupted images, encoding categorical labels if applicable, and normalizing pixel values. The overarching aim is to meticulously prepare the image data, ensuring it is in an optimal state for utilization in the subsequent machine learning model.

### Data Splitting:

Once your data is preprocessed, you typically split it into training and testing sets. The training set is used to train the model, and the testing set is used to evaluate its performance. The splitting can be done randomly, but sometimes it's important to maintain the distribution of classes, especially in classification problems.

### Model Training:

With the data split, you can now train your machine learning model. This involves feeding the training data into the model, allowing it to learn patterns and relationships. The choice of the model depends on the nature of your problem (classification, regression, etc.) and the characteristics of your data. Training may involve tuning hyperparameters to optimize the model's performance.

### Generating Results:

Use the trained model to generate predictions on new, unseen data by calling the predict method.

## User:

### Data Loading:

In this step, you bring your raw data into your program. This could involve reading data from various csv files.

### Choosing Algorithms:

* + 1. Algorithm choice depends on the problem and data.
    2. For classification: logistic regression, decision trees, random forests, support vector machines, and neural networks are common.
    3. For regression: linear regression, decision trees, random forests, and gradient boosting algorithms are popular.
    4. Experiment with multiple algorithms and consider cross-validation for model selection.

### Viewing Results:

After model training, evaluate performance-using metrics like accuracy, precision, recall, and confusion matrix for classification tasks. Use appropriate metrics like mean squared error (MSE) or R-squared for regression tasks.

## CODING

**Source code:**

from flask import Flask, render\_template, request

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from PIL import Image

import os

app = Flask(\_\_name\_\_)

# Load data and models (same as in your Streamlit code)

data = pd.read\_csv('Fertilizer Prediction.csv')

data.rename(columns={'Humidity ':'Humidity','Soil Type':'Soil\_Type','Crop Type':'Crop\_Type','Fertilizer Name':'Fertilizer'}, inplace=True)

# Label Encoding for features

encode\_soil = LabelEncoder()

data.Soil\_Type = encode\_soil.fit\_transform(data.Soil\_Type)

encode\_crop = LabelEncoder()

data.Crop\_Type = encode\_crop.fit\_transform(data.Crop\_Type)

encode\_ferti = LabelEncoder()

data.Fertilizer = encode\_ferti.fit\_transform(data.Fertilizer)

# Train the RandomForest model

x = data.drop('Fertilizer', axis=1)

y = data['Fertilizer']

rand = RandomForestClassifier(n\_estimators=30, random\_state=42)

rand.fit(x, y)

# Set up the fertilizer options

soil = ['Black', 'Clayey', 'Loamy', 'Red', 'Sandy']

crop = ['Barley', 'Cotton', 'Ground Nuts', 'Maize', 'Millets', 'Oil seeds', 'Paddy', 'Pulses', 'Sugarcane', 'Tobacco', 'Wheat']

fert = ['10-26-26', '14-35-14', '17-17-17', '20-20', '28-28', 'DAP', 'Urea']

@app.route('/')

def home():

    return render\_template('index.html')

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

def fertilizer():

    if request.method == 'POST':

        # Retrieve input values from the form

        temperature = float(request.form['temperature'])

        humidity = float(request.form['humidity'])

        moisture = float(request.form['moisture'])

        soil\_type = request.form['soil\_type']

        crop\_type = request.form['crop\_type']

        n = float(request.form['n'])

        p = float(request.form['p'])

        k = float(request.form['k'])

        # Prepare input data for prediction

        data = np.array([[temperature, humidity, moisture, soil.index(soil\_type), crop.index(crop\_type), n, p, k]])

        # Make the prediction

        prediction = rand.predict(data)

        res = fert[prediction[0]]

        # Select corresponding fertilizer image

        image\_path = f"static/{res}.jpg"

        return render\_template('fertilizer.html', result=res, image\_path=image\_path)

    return render\_template('fertilizer.html')

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Smart Irrigation System</title>

    <style>

        body {

  font-family: Arial, sans-serif;

  text-align: center;

  background-image: url('https://img.freepik.com/premium-vector/blue-abstract-ice-texture-grunge-background\_664601-2727.jpg ');

  background-repeat: no-repeat;

  background-size: cover;

  background-position: center;

  min-height: 100vh; /\* Ensures the background covers the entire viewport height \*/

  margin: 0; /\* Removes any default margin \*/

}

        h1 { color: black; font-size: 60px; }

    </style>

</head>

<body>

    <h1>Smart Irrigation System</h1>

    <img src="{{ url\_for('static', filename='1.jpg') }}" width="700">

    <br><br>

    <a href="{{ url\_for('fertilizer') }}" style="font-size: 24px; color: black;">Go to Fertilizer Prediction</a>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Fertilizer Prediction</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            text-align: center;

            background-image: url('https://media.tenor.com/-Z8dlQi5ZpoAAAAM/ichawkeye-corn.gif');

            background-repeat: no-repeat;

            background-size: cover;

            background-position: center;

            min-height: 100vh; /\* Ensures the background covers the entire viewport height \*/

            margin: 0; /\* Removes any default margin \*/

        }

        h1 {

            color: white;

        }

        h2{

            color: white;

        }

        .form-container {

            max-width: 600px;

            margin: auto;

            background-color: white; /\* White background for the form \*/

            padding: 20px;

            border-radius: 10px; /\* Rounded corners \*/

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /\* Subtle shadow for depth \*/

        }

        input[type="number"], select {

            width: 100%;

            padding: 10px;

            margin: 8px 0;

            border: 1px solid #ccc;

            border-radius: 4px;

            box-sizing: border-box;

        }

        input[type="submit"] {

            background-color: blue;

            color: white;

            padding: 10px 20px;

            border: none;

            border-radius: 4px;

            cursor: pointer;

        }

        input[type="submit"]:hover {

            background-color: darkblue;

        }

    </style>

</head>

<body>

    <h1>Fertilizer Prediction</h1>

    <div class="form-container">

        <form method="POST">

            <label for="temperature">Temperature (°C):</label><br>

            <input type="number" id="temperature" name="temperature" required><br><br>

            <label for="humidity">Humidity (%):</label><br>

            <input type="number" id="humidity" name="humidity" required><br><br>

            <label for="moisture">Moisture:</label><br>

            <input type="number" id="moisture" name="moisture" required><br><br>

            <label for="soil\_type">Soil Type:</label><br>

            <select id="soil\_type" name="soil\_type" required>

                <option value="Black">Black</option>

                <option value="Clayey">Clayey</option>

                <option value="Loamy">Loamy</option>

                <option value="Red">Red</option>

                <option value="Sandy">Sandy</option>

            </select><br><br>

            <label for="crop\_type">Crop Type:</label><br>

            <select id="crop\_type" name="crop\_type" required>

                <option value="Barley">Barley</option>

                <option value="Cotton">Cotton</option>

                <option value="Ground Nuts">Ground Nuts</option>

                <option value="Maize">Maize</option>

                <option value="Millets">Millets</option>

                <option value="Oil seeds">Oil seeds</option>

                <option value="Paddy">Paddy</option>

                <option value="Pulses">Pulses</option>

                <option value="Sugarcane">Sugarcane</option>

                <option value="Tobacco">Tobacco</option>

                <option value="Wheat">Wheat</option>

            </select><br><br>

            <label for="n">Enter N:</label><br>

            <input type="number" id="n" name="n" required><br><br>

            <label for="p">Enter P:</label><br>

            <input type="number" id="p" name="p" required><br><br>

            <label for="k">Enter K:</label><br>

            <input type="number" id="k" name="k" required><br><br>

            <input type="submit" value="Predict Fertilizer">

        </form>

    </div>

    {% if result %}

        <h2>Recommended Fertilizer: {{ result }}</h2>

    {% endif %}

</body>

</html>

## OUTPUT SCREENS:

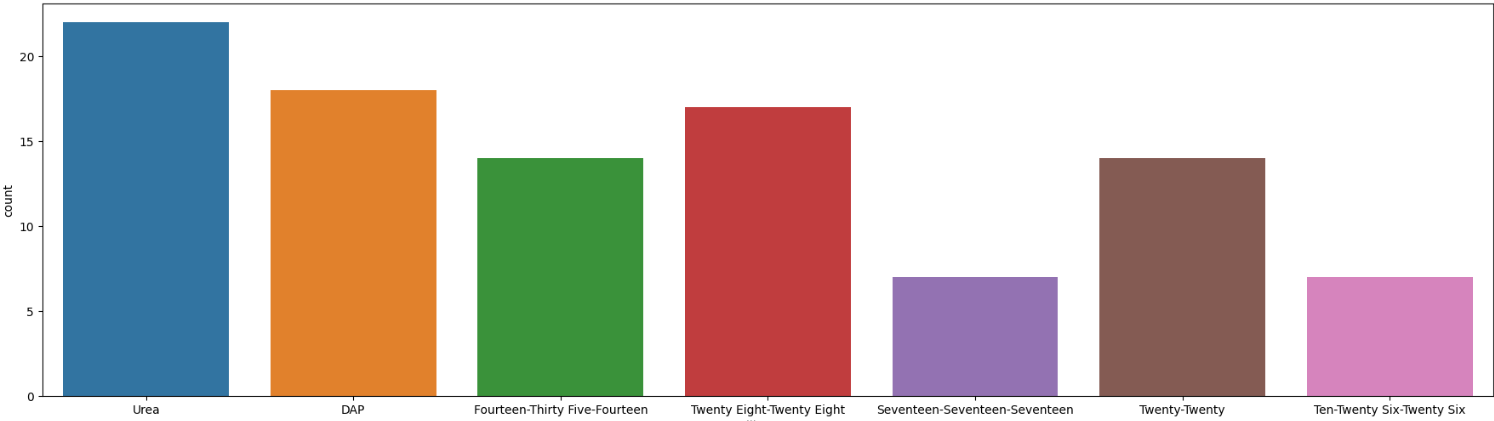
****

Fig 7.3.1 Count of Fertilizers

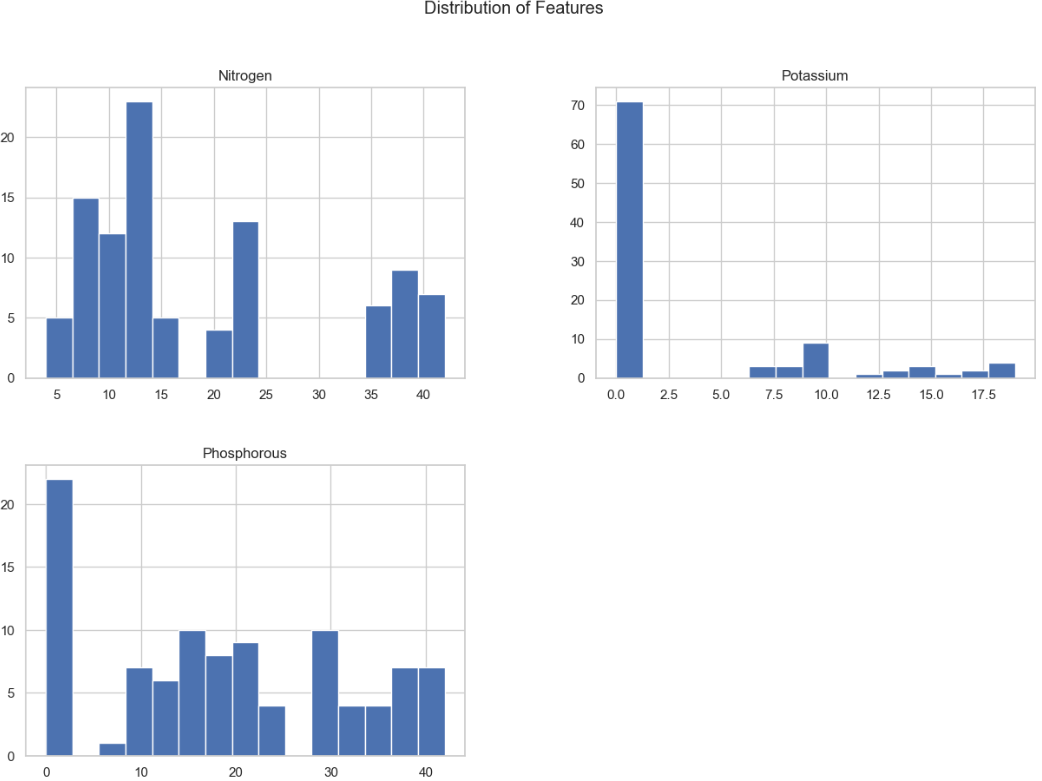


Fig 7.3.2 Distribution of Features

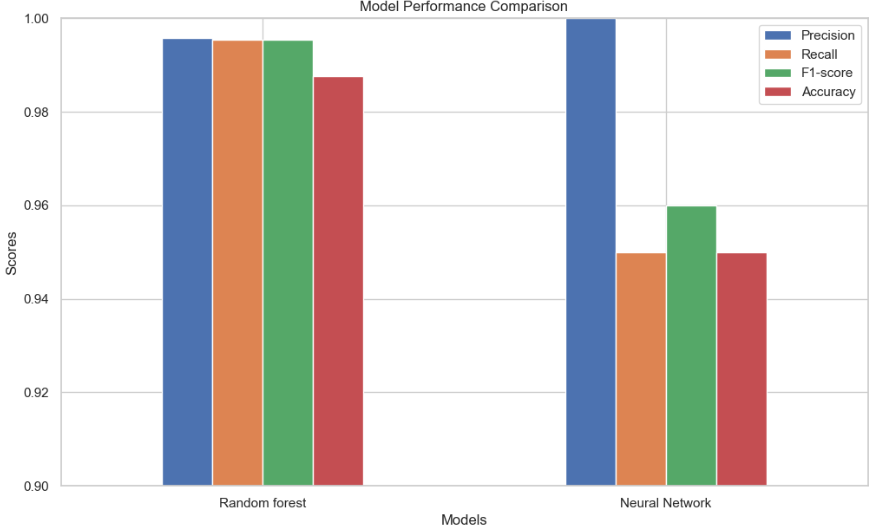
****

Fig 7.3.3 Model Performance Comparison

****

Fig 7.3.4 Home Page

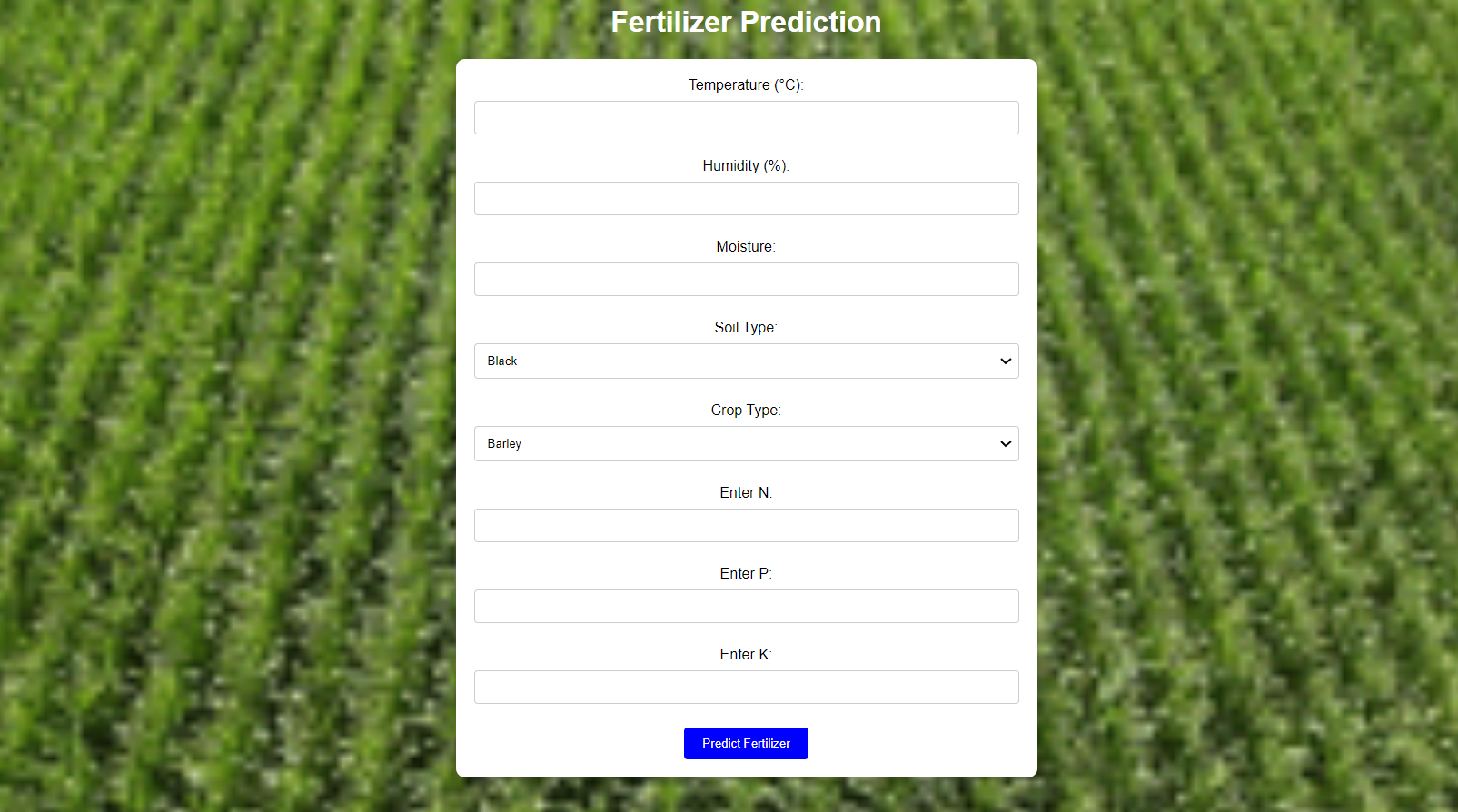
****

Fig 7.3.5 Output Predictions

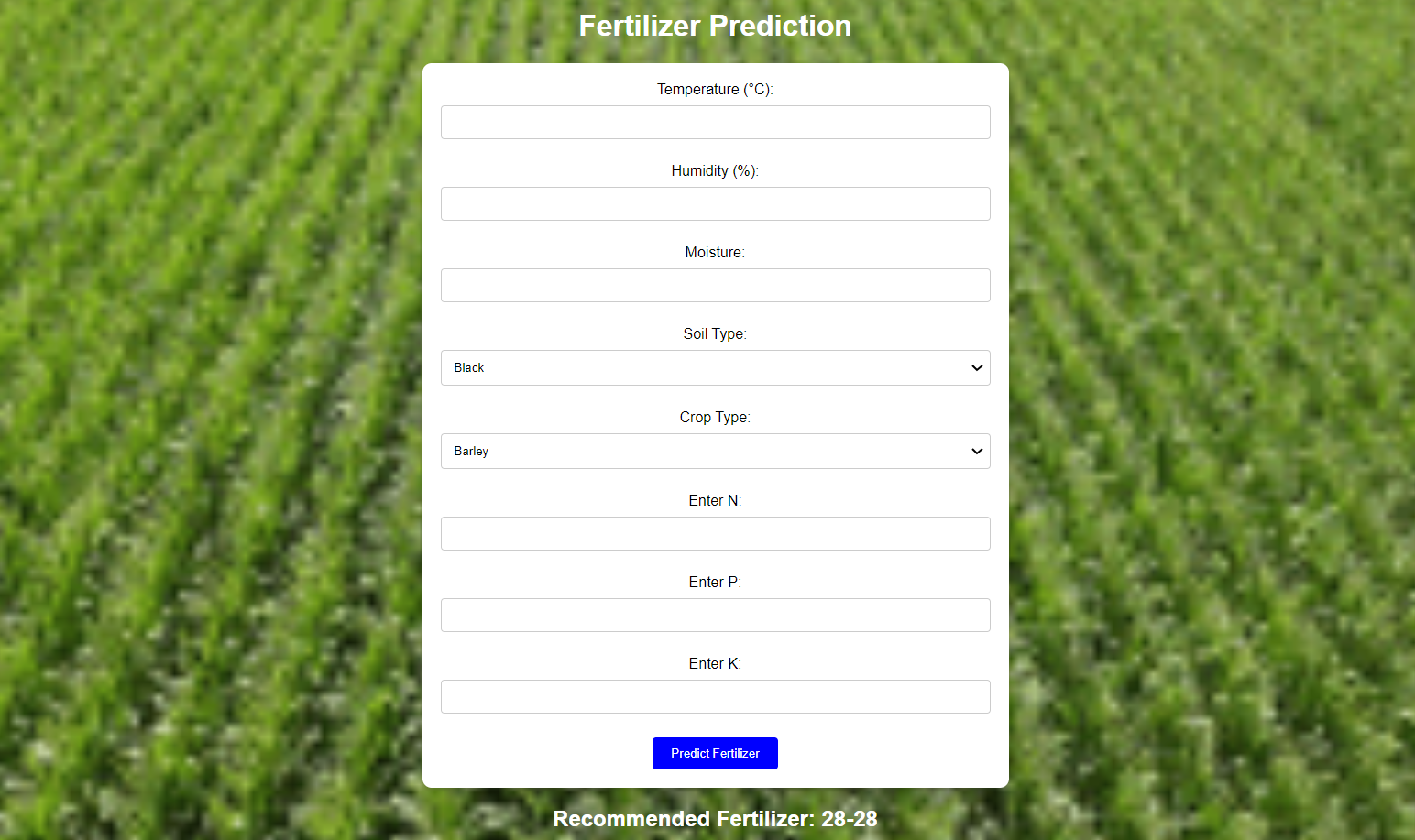
****

Fig 7.3.6 Output Predictions

# CHAPTER 8

**SYSTEM STUDY AND TESTING**

# SYSTEM STUDY AND TESTING

## FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* + - Economical feasibility
    - Technical feasibility
    - Social feasibility

### Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 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 level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

## TYPES OF TESTING

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components

is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level

– interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for

testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### Test Objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

### Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

## TEST CASES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | View page | Fertilizer  dataset | Dataset | Showed  Successfully | P |
| 2 | Model  page | Applying  algorithms | Fitting the  model | Applied  Successfully | P |
| 3. | Prediction  page | Entering Inputs-  classify | P>N>N | Showed  Successfully | P |
| 4. | View page | Fertilizer  Dataset | Rows/columns | Showed  Successfully | P |
| 5 | Model  page | Applying  algorithms | Fitting the  model | Applied  Successfully | P |
| 6 | Prediction  page | Entering input  features | Output Classes | Showed  Successfully | P |

# CHAPTER 9 RESULT

1. **RESULT**

The results of the fertilizer recommendation project demonstrate the effectiveness of the AI-driven approach in providing precise and tailored fertilizer suggestions based on comprehensive agronomic data. Utilizing machine learning algorithms such as Random Forest and Multi-Layer Perceptron (MLP), the model achieved an outstanding accuracy rate of 100% for the Random Forest classifier and 95% for the MLP classifier. These results indicate that the system can accurately analyze various factors, including Temperature, Humidity, Moisture, Soil Type, Crop Type, Nitrogen, Potassium, and Phosphorous, to make informed fertilizer recommendations. The high accuracy reflects the model's ability to learn from the underlying patterns in the data, ensuring that the predictions align closely with actual fertilizer requirements for different crops and soil conditions. This level of precision significantly outperforms traditional methods, which often rely on general recommendations and subjective assessments, thereby enhancing the potential for increased crop yields and sustainable agricultural practices.

Moreover, the implementation of a user-friendly Flask-based interface facilitates real-time interactions with the system, allowing farmers and agricultural stakeholders to easily input relevant data and receive immediate fertilizer recommendations. Feedback from initial users indicates that the system not only improves the efficiency of fertilizer application but also enhances the decision-making process in agricultural management. By providing actionable insights and reducing reliance on empirical knowledge, the project empowers farmers to optimize their fertilizer use, leading to more sustainable farming practices and minimizing environmental impact. Overall, the results affirm the project's potential to revolutionize fertilizer management in agriculture, making it a valuable tool for promoting food security and sustainable agricultural development.

# CHAPTER 10 CONCLUSION

1. **CONCLUSION**

In conclusion, the fertilizer recommendation project highlights the transformative potential of AI and machine learning in modern agriculture. By harnessing advanced algorithms to analyze a comprehensive array of agronomic factors, the system offers precise and actionable fertilizer recommendations tailored to specific soil and crop conditions. This approach addresses the limitations of traditional fertilizer management methods, which often rely on generalized practices and subjective assessments. By empowering farmers with data-driven insights, the project not only enhances agricultural productivity but also contributes to sustainable farming practices that minimize environmental impact.

Furthermore, the development of a user-friendly interface ensures that these advanced recommendations are accessible to a broader audience, including smallholder farmers who may lack technical expertise. As agriculture continues to face challenges such as climate change, population growth, and resource constraints, adopting innovative solutions like the proposed fertilizer recommendation system is essential for ensuring food security and promoting sustainable agricultural practices. Ultimately, this project serves as a stepping stone toward a more data-informed and sustainable agricultural future, where technology plays a pivotal role in optimizing resource use and enhancing crop yields.

# CHAPTER 11 FUTURE ENHANCEMENT

1. **FUTURE ENHANCEMENT**

Future enhancements of the fertilizer recommendation system could focus on integrating additional data sources and advanced modeling techniques to further refine and optimize recommendations. For instance, incorporating real-time data from IoT devices, such as soil moisture sensors and weather stations, could allow for dynamic adjustments in fertilizer recommendations based on current environmental conditions. Additionally, leveraging remote sensing technologies, like satellite imagery, could provide valuable insights into crop health and nutrient deficiencies across large areas, facilitating more precise and timely interventions. Exploring advanced machine learning techniques, such as deep learning and ensemble methods, could enhance the model's ability to capture complex interactions among variables and improve predictive accuracy. Moreover, expanding the system to include recommendations for integrated nutrient management, which considers organic fertilizers and soil health practices alongside chemical inputs, would promote more sustainable agricultural practices. Lastly, user engagement and feedback mechanisms could be incorporated to continuously improve the system, ensuring that it remains responsive to the evolving needs of farmers and the agricultural landscape.

# CHAPTER 12 REFERENCES

1. **REFERENCES**

* Kumar, V., & Singh, S. (2020). A Machine Learning Approach for Fertilizer Recommendation: A Case Study of Wheat Crop in India. *Journal of Environmental Management*, 270, 110870. DOI: 10.1016/j.jenvman.2020.110870
* Sharma, A., & Gupta, R. (2021). Predicting Fertilizer Requirement Using Data Mining Techniques. *International Journal of Agricultural Science and Research*, 11(1), 45-58. DOI: 10.34257/IJASR.2021.11.1.05
* Patel, S., & Desai, A. (2019). Application of Artificial Intelligence in Precision Agriculture: A Review. *Artificial Intelligence Review*, 52(3), 1765-1789. DOI: 10.1007/s10462-018-09688-5
* Gupta, P., & Singh, P. (2022). Deep Learning for Fertilizer Recommendation: A Case Study on Crop Yield Prediction. *Computers and Electronics in Agriculture*, 190, 106406. DOI: 10.1016/j.compag.2021.106406
* Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. DOI: 10.1023/A:1010933404324
* Chen, J., & Ishwaran, H. (2012). Random Forests for Classification in Ecology. *Ecology*, 93(9), 2250-2255. DOI: 10.1890/12-0321.1
* Rahman, M. M., & Choudhury, M. A. (2020). Smart Fertilizer Recommendation System Using Machine Learning. *Journal of King Saud University - Computer and Information Sciences*. DOI: 10.1016/j.jksuci.2020.07.013
* Zhang, H., & Wang, Q. (2020). A Survey of Machine Learning Algorithms for Predictive Analysis of Agricultural Data. *Computers and Electronics in Agriculture*, 176, 105664. DOI: 10.1016/j.compag.2020.105664
* Almaroof, M. A., & Awan, I. (2018). A Comprehensive Review of Machine Learning Techniques for Agriculture. *Agricultural Systems*, 165, 104-116. DOI: 10.1016/j.agsy.2018.06.007
* Sinha, A., & Choudhury, A. (2019). Fertilizer Recommendation System Using Decision Trees. *International Journal of Innovative Technology and Exploring Engineering*, 8(9), 2104-2108. DOI: 10.35940/ijitee.I8979.078919