A Project Report On

AGRICULTURE YIELD PREDICTION SYSTEM USING MACHINE LEARNING

Major project submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
(2021-2025)
BY

T. ARCHANA	21241A12J7
B. SHIVANI	21241A12D3
E. REENA	21241A12E4

Under the Esteemed Guidance
of

J. ALEKHYA

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

HYDERABAD

2023-24

CERTIFICATE

This is to certify that it is a bonafide record of Major Project work entitled **"AGRICULTURE YIELD PREDICTION SYSTEM USING MACHINE LEARNING"** done by **T. ARCHANA (21241A12J7), B. SHIVANI (21241A12D3), E. REENA (21241A12E4)** of **B.Tech** in the Department of Information of Technology, **Gokaraju Rangaraju Institute of Engineering and Technology** during the period 2021-2025 in the partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET, Hyderabad.

J. Alekhya
Assistant Professor
(Internal Guide)

Dr. Y J Nagendra Kumar
Head of the Department

(Project External)

ACKNOWLEDGEMENT

We take the immense pleasure in expressing gratitude to our Internal guide**, J. Alekhya, Assistant Professor, Dept of IT**, GRIET. We express our sincere thanks for her encouragement, suggestions and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr. Y J Nagendra Kumar,** HOD IT,our Project Coordinators **Dr. N Rajasekhar, P. Gopala Krishna** and **P. Bharathi** for their constant support during the project.

We express our sincere thanks to **Dr. Jandhyala N Murthy,** Director, GRIET**,** and **Dr. J. Praveen,** Principal, GRIET**,** for providing us the conductive environment for carrying through our academic schedules and project with ease.

We also take this opportunity to convey our sincere thanks to the teaching and non-teaching staff of GRIET College, Hyderabad.

Name: T. Archana
Email: archanathota2003@gmail.com
Contact No. 6301506022

Name: B. Shivani
Email: bandishivaniit2021@gmail.com
Contact No. 8341495388

Name: E. Reena
Email: reenaerukulla19@gmail.com
Contact No. 9014083245

DECLARATION

This is to certify that the major-project entitled "**AGRICULTURE YIELD PREDICTION SYSTEM USING MACHINE LEARNING"** is a bonafide work done by us in partial fulfillment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites, books and paper publications are mentioned in the Bibliography.

This work was not submitted earlier at any other University or Institute for the award of any degree.

T. ARCHANA **21241A12J7**

B. SHIVANI **21241A12D3**

E. REENA **21241A12E4**

TABLE OF CONTENTS

11 LIST OF DIAGRAMS

ABSTRACT

India, being an agrarian economy, heavily relies on agriculture as the backbone of its economic structure. The majority of its population is engaged in agricultural activities, and the sector significantly contributes to the GDP. However, modern agriculture faces critical challenges, primarily due to the effects of climate change and environmental uncertainties. These issues impact crop productivity and thereby influence the livelihoods of farmers and the overall economy. Addressing these challenges requires innovative and sustainable solutions, and machine learning (ML) offers a promising avenue to enhance decision-making and optimize outcomes in the agricultural sector.

Crop Yield Prediction, a crucial application of machine learning, involves analyzing historical data such as weather parameters, soil characteristics, and past crop yields to estimate future productivity. This approach not only provides farmers with insights about the expected yield of their crops but also helps them plan better and mitigate potential risks. The Random Forest algorithm, a robust supervised machine learning method, is particularly effective for such predictions. It operates by constructing multiple decision trees during training and outputs the average of predictions, ensuring high accuracy and reliability. This algorithm is well-suited for handling complex datasets with multiple variables, making it ideal for agriculture-related predictions.

The proposed project focuses on building a web-based application that integrates this predictive capability to benefit farmers. By inputting key attributes like crop type, soil pH, and rainfall data, the application predicts crop yield and suggests alternative high-yield crops for given conditions. Such recommendations empower farmers to make informed decisions, maximize productivity, and adapt to changing environmental conditions. This application not only aids in resource optimization but also contributes to food security and economic stability by fostering more sustainable agricultural practices. Through the adoption of machine learning, this project paves the way for smarter farming solutions and a resilient agricultural sector.

Keywords: Agriculture yield prediction system, Random forest

Domain: Machine learning

1. INTRODUCTION

1.1 Introduction to Project

India is predominantly an agrarian country, with its economy heavily reliant on agriculture and crop productivity. Agriculture serves as the backbone of numerous businesses and industries, directly or indirectly influencing the nation's economic growth. Given its significance, any challenges or disruptions in the agricultural sector can have widespread implications. However, agriculture today faces severe challenges due to climate change and environmental fluctuations, which have become major threats to sustainable farming practices and crop productivity.

In this context, advancements in technology, particularly in the field of machine learning (ML), have emerged as promising tools to address these challenges. Machine learning, with its ability to analyze large datasets and uncover patterns, provides practical and effective solutions for agricultural problems. Among its many applications, crop yield prediction is particularly impactful. By leveraging historical data such as weather conditions, soil parameters, and past crop yields, ML models can accurately predict the yield of various crops, helping farmers make informed decisions.

The Random Forest algorithm, a powerful and popular supervised machine learning technique, is particularly suited for this task. Known for its robustness and high accuracy, the algorithm analyzes complex datasets to identify trends and provide reliable predictions. This project aims to utilize the Random Forest algorithm to predict crop yields, enabling farmers to anticipate outcomes before planting their crops. Such predictions can guide resource allocation and reduce the risks associated with agriculture.

Beyond merely predicting yields, the project extends its scope to recommend high-yielding crops based on specific conditions such as soil pH, rainfall, and other environmental factors. By combining prediction with recommendation, the solution not only empowers farmers with foresight but also optimizes their cultivation strategies. This dual approach ensures better resource utilization, improved productivity, and higher profitability for farmers.

In an increasingly data-driven healthcare environment, the Medical Prescription OCR system holds the potential to improve patient safety, enhance the quality of healthcare delivery, and streamline administrative processes. This project not only showcases technical expertise but also represents a meaningful contribution to the broader field of healthcare technology, with far-reaching implications for patient well-being and the efficiency of healthcare services.

1.2 Motivation

Agriculture forms the foundation of every economy, and in a country like India, with its rapidly growing population, advancements in the agricultural sector are crucial to meet increasing food demands. Historically, agriculture has been a central part of Indian culture, with ancient practices centered on cultivating crops sustainably to meet local needs. However, with the advent of innovative technologies and hybrid techniques, traditional agricultural methods have been overshadowed, leading to challenges such as soil degradation, reduced biodiversity, and reliance on artificial products, which compromise health and sustainability. Modern farmers often lack awareness of optimal cultivation practices, such as planting the right crops at the right time and place, which, combined with shifting seasonal climates, have further stressed natural resources like soil, water, and air. These changes have resulted in food insecurity and environmental concerns. Despite these challenges, there are numerous ways to enhance crop yield and quality through informed approaches, leveraging modern tools and techniques that address critical factors like weather, temperature, and soil conditions.

1.3 Objective of the project

This project is designed to address the challenges faced by farmers in determining the best crop to grow under specific conditions. It employs machine learning techniques to predict crop yields based on various attributes such as crop type, soil pH levels, rainfall patterns, temperature, and other environmental factors. By analyzing historical and real-time data, the project identifies trends and relationships that are not immediately evident, providing farmers with actionable insights. The predictions help farmers make informed decisions about which crops will yield the most under prevailing conditions, optimizing both productivity and profitability.

The end result of this initiative is a web application that integrates machine learning with agriculture. This application not only predicts crop yields but also recommends high-yielding crops suitable for a given set of conditions. By offering data-driven recommendations, it bridges the gap between traditional farming practices and modern technology. This tool empowers farmers with knowledge, helping them navigate the uncertainties of climate and environmental changes. As a result, it fosters sustainable agricultural practices, reduces risks, and enhances overall productivity, contributing significantly to the growth of the agricultural sector.

Random Forest Algorithm:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points.

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

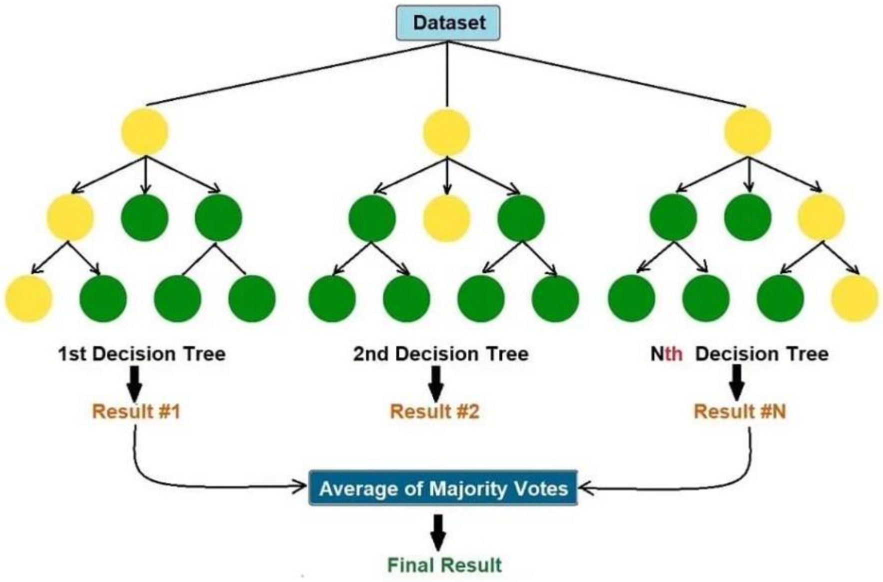


Figure:1 Random Forest Algorithm

Decision Tree Algorithm:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes 18 represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

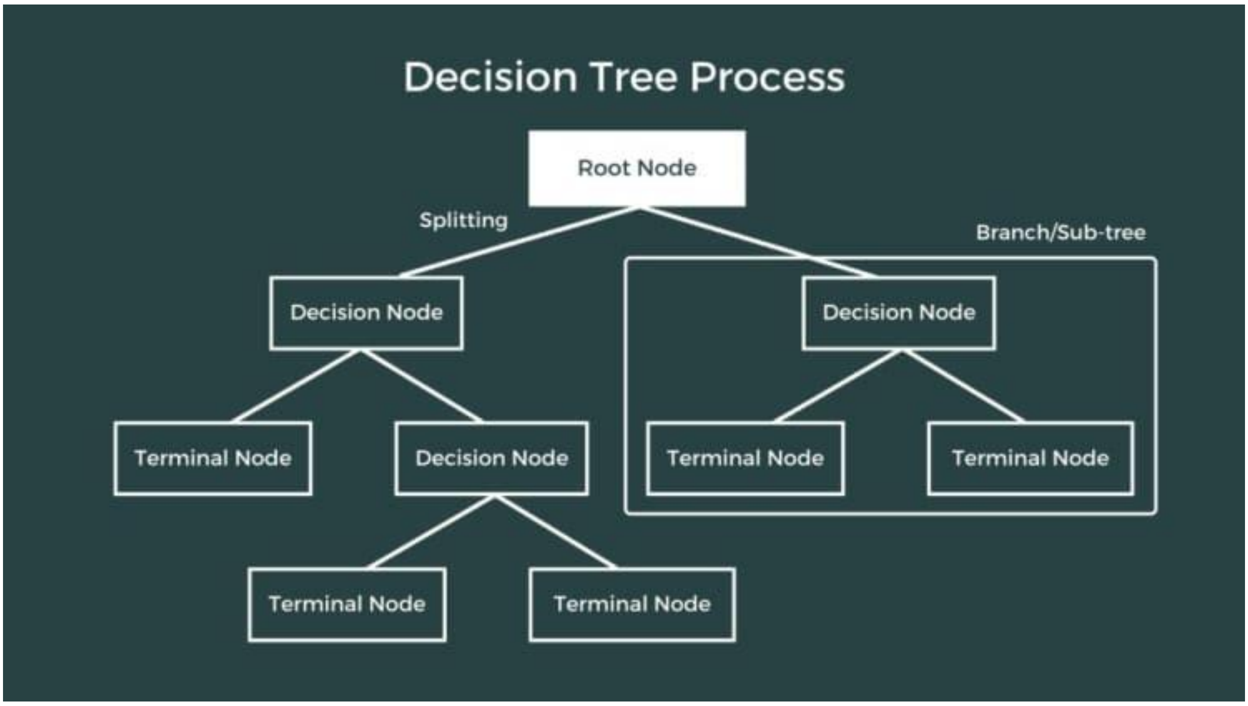


Figure:2 Decision Tree

1.4 Existing System

The current crop yield prediction systems utilize algorithms such as Logistic Regression, Naïve Bayes, and Random Forest, offering basic yield predictions based on limited parameters like weather and soil conditions. While these models provide some insight, they lack a crop recommendation feature, which could guide farmers in selecting high-yield crops suited to their environmental conditions. Multiple Linear Regression is also employed for yield prediction by considering factors like weather, soil quality, and management practices, but it fails to address the need for actionable crop recommendations. The accuracy of these systems is further limited by the exclusion of critical parameters such as advanced climate data, soil fertility, pest resistance, and economic factors. To overcome these shortcomings, integrating real-time data, advanced machine learning models, and user-friendly interfaces can create a more robust system that not only predicts yields but also recommends suitable crops tailored to specific conditions.

Limitations of Existing System:

1. **Restricted Parameters Considered:** Current models rely on basic weather and soil conditions, neglecting other crucial factors like pest resistance and economic influences.
2. **No Crop Recommendation Feature:** The systems lack a functionality to recommend suitable and high-yield crops tailored to specific environmental and soil conditions.

3. **Exclusion of Critical Factors:** Important parameters, such as advanced climate data, soil fertility, historical crop performance, and market demand, are not accounted for in predictions.
4. **No Real-Time Data Integration:** IoT devices and sensors are not utilized to provide real-time data on soil moisture, nutrient levels, and microclimatic conditions, limiting dynamic predictions.
5. **Limited User Accessibility:** The systems do not offer user-friendly interfaces or multilingual support, making them less practical and accessible for diverse farming communities.
6. **Suboptimal Prediction Accuracy:** The exclusion of advanced attributes and modern machine learning techniques, such as ensemble or deep learning models, leads to reduced accuracy and reliability.

1.5 Proposed System

The proposed system is a web-based application designed to predict crop yields accurately while also recommending high-yield crops tailored to specific environmental and soil conditions. This innovative system bridges the gap in current agricultural practices by combining prediction and recommendation functionalities, enabling farmers to make informed decisions that enhance productivity and profitability. By incorporating diverse parameters such as temperature, rainfall, season, nitrogen levels, and area under cultivation, the system ensures adaptability to various regions and farming scenarios. Leveraging the Random Forest Regressor algorithm, the system provides precise, reliable predictions and recommendations, supported by an intuitive interface that makes it accessible and practical for a broad range of users.

Key features of the proposed system include:

1. **Dual Functionality:** Predicts crop yields and recommends high-yield crops based on specific conditions.
2. **Incorporation of Diverse Parameters:** Considers critical factors like temperature, rainfall, soil conditions, and nitrogen content.
3. **Advanced Algorithm:** Utilizes the Random Forest Regressor for precise and reliable predictions.
4. **User-Friendly Interface:** Offers an intuitive design with easy data input and visual outputs such as charts and recommendations.
5. **Broad Applicability:** Adaptable to various regions and farming scenarios, ensuring relevance for diverse users.

2. REQUIREMENT ENGINEERING

2.1 Hardware Requirements

- Processor –Intel core i3 and above
- Memory – 4GB RAM and above
- Input devices – Keyboard Mouse
- Internet

2.2 Software Requirements

- Operating System - Any OS capable of running Python and web browsers (Mac, Windows, Linux).
- Front-end - HTML, CSS.
- Libraries – pandas, scikit-learn.
- Languages - Python 3.x.
- Tools Required - Jupyter Notebook

3. LITERATURE SURVEY

[1] Title: A Review on Data Mining Techniques for Fertilizer Recommendation, 2018.

Authors : Jignasha M. Jethva, Nikhil Gondaliya, Vinita Shah

To keep up nutrition levels in the soil in case of deficiency, fertilizers are added to soil. The standard issue existing among the Indian agriculturists choose approximate amount of fertilizers and add them manually. Excess or deficient extension of fertilizers can harm the plants life and reduce the yield. This paper gives overview of various data mining frameworks used on cultivating soil dataset for fertilizer recommendation.

[2] Title: A Survey on Data Mining Techniques in Agriculture, 2015.

Authors : M.C.S.Geetha

Agriculture is the most critical application area especially in the developing nations like India .Use of information technology in agriculture can change the situation of decision making and farmers can yield in better way.. This paper integrates the work of several authors in a single place so it is valuable for specialists to get data of current situation of data mining systems and applications in context to farming field.

[3] Title : AgroNutri Android Application, 2016.

Authors : S. Srija, R. Geetha Chanda, S.Lavanya, Dr. M. Kalpana Ph.D

This paper communicates the idea regarding the making of AgroNutri an android application that helps in conveying the harvest particular fertilizer amount to be applied. The idea is to calculate the measure of NPK composts to be applied depend on the blanked proposal of the crop of interest. This application works depend on the product chosen by the farmer and that is taken as input, thus providing the farmers. The future scope of the AgroNutri is that GPRS can be included so that according to location nutrients are suggested.

[4] Title: Machine Learning: Applications in Indian Agriculture, 2016.

Authors: Karandeep Kaur

Agriculture is a field that has been lacking from adaption of technologies and their advancements. Indian agriculturists should be up to the mark with the universal procedures. Machine learning is a native concept that can be applied to every field on all inputs and outputs. It has effectively settled its ability over ordinary calculations of software engineering and measurements. Machine learning calculations have improved the exactness of artificial intelligence machines including sensor based frameworks utilized in accuracy farming. This paper has evaluated the different uses of machine learning in the farming area. It additionally gives a knowledge into the inconveniences looked by Indian farmers and how they can be resolved using these procedures.

[5] Title: Impacts of population growth, economic development, and technical change on global food production and consumption, 2011.

Author: Uwe A. Schneider a,⇑, Petr Havlik b, Erwin Schmid c, Hugo Valin b, Aline Mosnier b,c, Michael Obersteiner b, Hannes Bottcher b, Rastislav Skalsky´ d, Juraj Balkovicˇ d, Timm Sauer a, Steffen Fritz b

Throughout the following decades humanity will request more food from less land and water assets. This investigation evaluates the food production effects of four elective advancement situations from the Millennium Ecosystem Assessment and the Special Report on Emission Scenarios. partialy and jointly considered are land and water supply impacts from population development, and specialized change, and forests and agriculture demand request shifts from population development and economic improvement. The income impacts on nourishment request are registered with dynamic flexibilities. Worldwide farming area increments by up to 14% somewhere in the range of 2010 and 2030.Deforestation restrictions strongly impact the price of land and water resources but have little consequences for the global level of food production and food prices. While projected income changes have the highest partial impact on per capita food consumption levels, population growth leads to the highest increase in total food production. The impact of technical change is amplified or mitigated by adaptations of land management intensities.

[6] Title: Brief history of agricultural systems modelling, 2016.

Author: James W. Jones a,⁎, John M. Antle b, Bruno O. Basso c, Kenneth J. Boote a, Richard T. Conant d, Ian Foster e, H. Charles J. Godfray f, Mario Herrero g, Richard E. Howitt h, Sander Jansseni, Brian A. Keating g, Rafael Munoz-Carpena a, Cheryl H. Porter a, Cynthia Rosenzweig j, Tim R.Wheeler k

Rural frameworks science creates information that enables analysts to consider complex issues or take educated farming choices. The rich history of this science represents the decent variety of frameworks and scales over which they work and have been contemplated. Demonstrating, a basic apparatus in agrarian frameworks science, has been expert by researchers from an extensive variety of controls, who have contributed ideas and instruments over six decades. As agrarian researchers currently consider the "people to come" models, information, and learning items expected to meet the inexorably mind boggling frameworks issues looked by society, it is vital to check out this history and its exercises to guarantee that we stay away from re-innovation and endeavor to think about all elements of related difficulties. To this end, we outline here the historical backdrop of rural frameworks demonstrating and distinguish exercises discovered that can help control the structure and advancement of up and coming age of farming framework apparatuses and techniques. Various past occasions joined with generally innovative advancement in different fields have unequivocally added to the development of farming framework demonstrating, including improvement of process-based bio-physical models of yields and domesticated animals, factual models dependent on verifiable perceptions, and financial streamlining and reproduction models at family unit and local to worldwide scales. Attributes of rural frameworks models have changed broadly relying upon the frameworks included, their scales, and the extensive variety of purposes that spurred their advancement and use by specialists in various controls. Late patterns in more extensive joint effort crosswise over establishments, crosswise over orders, and between people in general and private segments recommend that the stage is set for the significant advances in rural frameworks science that are required for the up and coming age of models, databases, learning items and choice emotionally supportive networks.

[7] Title: A Smart Agricultural Model by Integrating Iot, Mobile and Cloud-based Big Data Analytics, 2017.

Authors: S.Rajeswari, K.Suthendran, K.Rajkumar.

In the cultivating field, the system models play a significant role to the enhancement of the agro-normal and money related conditions. In the proportions of benefits of the field and farm examinations to give the information and to recognize fitting and fruitful organization practices. It can recognize the organization to arrive managers and transversely over reality as long as the required soil, the board, environment, and money related information. Decision Support Systems (DSSs) use to make the information for the vermin the board, develop the officials. These systems are not using the impelled strategies to process the data. Thusly, use the adroit system thoughts to take the decisions for the issue. It expects a crucial activity in the comprehension of agronomic results, and their use as decision sincerely steady systems for farmers is extending.

[8] Title: An Overview of Internet of Things and Data Analytics in Agriculture: Benefits and Challenges, 2018.

Authors: Olakunle Elijah, Tharek Abdul Rahman, Igbafe Orikumhi, Chee Yen Leow, Nour

Hindia.

A blueprint of Iot and DA in agriculture has been shown in this paper. A couple of zones related to the association of Iot in agribusiness have been discussed in detail. The investigation of composing exhibits that there are clusters of work advancing being produced of Iot development that can be used to increase operational efficiency and gainfulness of plant and creatures. The benefits of Iot and DA, and open troubles have been identified and inspected in this paper. Iot is depended upon to offer a couple of benefits to the agribusiness division. Regardless, there are up 'til now different issues to be steered to make it moderate for close to nothing and medium-scale farmers. The key issues are security and cost. It is typical that as contention increases in the cultivating part.

[9] Title: Circulation Mode Selection Based on Cost Analysis, 2017.

Authors: Xiurong Sun*, Jingshan Zhang, Chenglin Wang, Tao Zhang

If every farmer and each average production base will join their optimal conditions in making cooperatives, it will accomplish economies of scale. Furthermore, producers will have an all the more favourable position in the plans with downstream firms (shipper or retailer).Second, the main customers of wholesale market are not inhabitants nearby who buy small quantities products but lower distributors or retailers. More redesigned transportation mode respects intensive attempt of new agrarian things, which prompts bolster the movement of new chain joint logistics and strengthen resource utilize and made logistics advantage quality. Refresh everything considered agrarian things spread. By then, regard the examination of gigantic worth control of standard things and achieve the mind blowing control to stream process.

4. TECHNOLOGY

4.1 ABOUT PYTHON

Python's environment has evolved significantly, enhancing its capabilities for statistical analysis. It strikes a fine balance between scalability and elegance, placing a premium on efficiency and code readability. Python is renowned for its emphasis on program readability, featuring a straightforward syntax that is beginner-friendly and encourages concise code expression through indentation. Noteworthy aspects of this high-level language include dynamic system functions and automatic memory management.

Python has gained widespread popularity due to its versatility and robust ecosystem of libraries and frameworks. These tools extend its capabilities for a variety of applications, including data analysis, machine learning, web development, and automation. Libraries such as NumPy, Pandas, and Matplotlib make it particularly powerful for statistical analysis and data visualization, while frameworks like TensorFlow and PyTorch have revolutionized machine learning workflows. Python’s ability to integrate seamlessly with other programming languages and tools further enhances its adaptability, making it a preferred choice for both beginners and experienced developers across diverse fields.

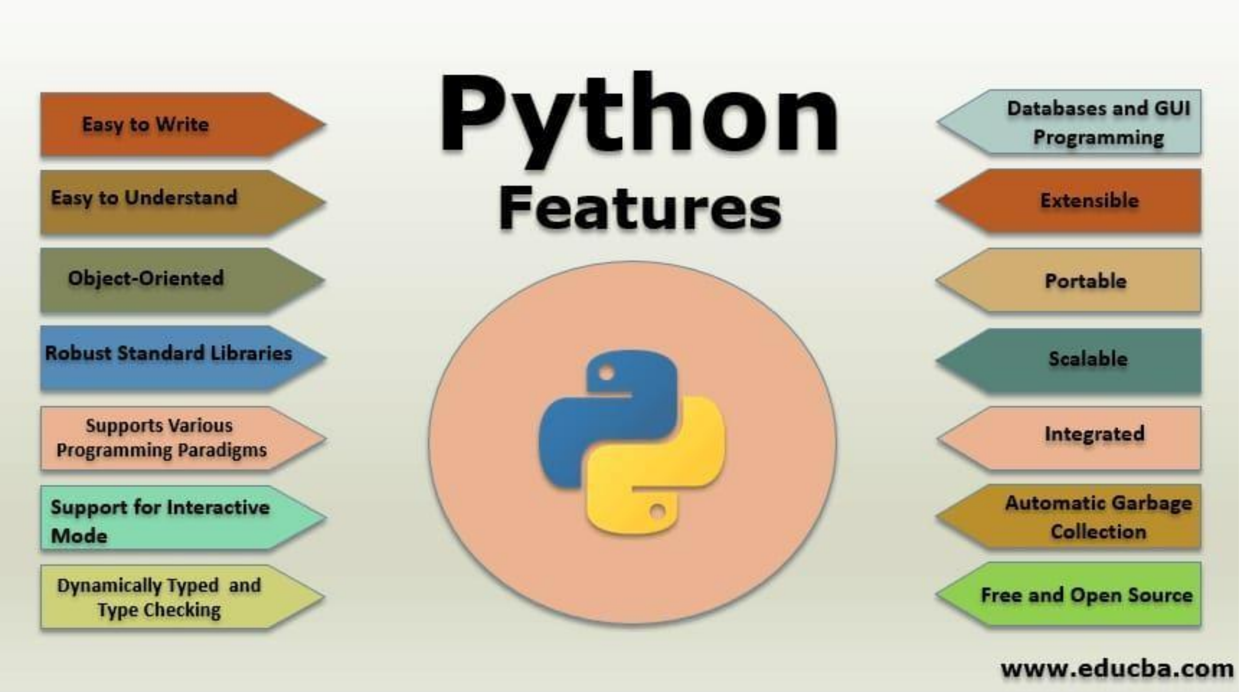


Figure:3 Features of Python

4.2 APPLICATIONS OF PYTHON

Python is used in many application domains. It makes its presence in every emerging field. It is the fastest-growing programming language and may be used to create any type of application.

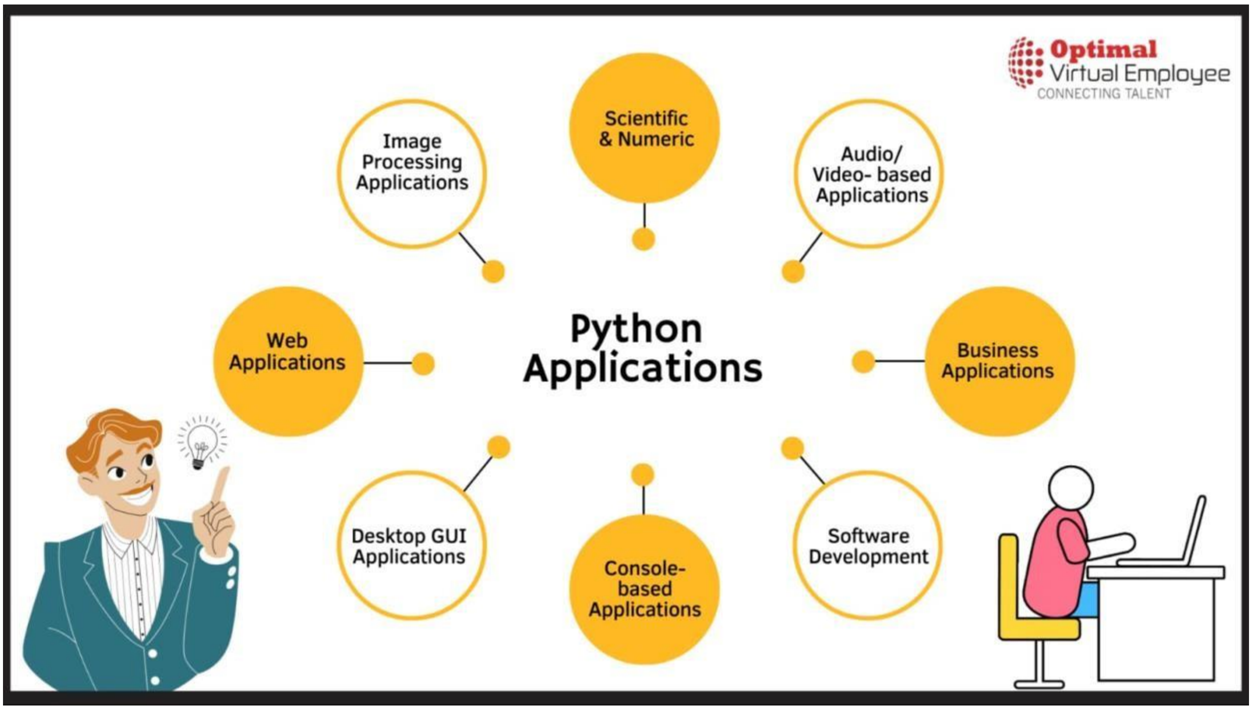


Figure:4 Applications of Python

Python is used in various fields:

- Web Applications. We can use Python to develop web applications. ...
- Desktop GUI Applications.
- Console-based Application.
- Software Development.
- Scientific and Numeric.
- Business Applications.
- Audio or Video-based Applications.
- Image processing.

4.3 PYTHON IS WIDELY USED IN MACHINE LEARNING

Python is one of the most widely used programming languages in the field of machine learning due to its flexibility, versatility, and open-source nature. It offers a wide array of tools and libraries specifically designed for mathematical computations, data preprocessing, and scientific operations, which are essential for building and deploying machine learning models. Libraries like NumPy, Pandas, and SciPy streamline data manipulation and analysis, while specialized frameworks such as TensorFlow, PyTorch, and Scikit-learn simplify the implementation of complex algorithms. Python's simple and intuitive syntax allows developers to focus more on solving problems rather than dealing with intricate coding structures, thereby reducing development time and effort. Furthermore, its extensive community support ensures that practitioners have access to a wealth of resources, tutorials, and prebuilt models, making it easier to experiment and innovate. These advantages make Python the go-to language for machine learning practitioners aiming for efficiency, scalability, and accuracy in their projects.

The major Python libraries used in machine learning are as follows:

4.3.1 PANDAS

Pandas is a Python library used for statistical analysis, data cleaning, exploration, and manipulation. Typically, datasets contain both useful and extraneous information. Pandas helps to make this data more readable and relevant.

4.3.2 NUMPY

NumPy is a Python library utilized for numerical data reading, cleaning, exploration, and manipulation. It provides powerful data structures for efficient computation with large arrays and matrices, making the data more accessible and manageable.

4.3.3 SCIKIT-LEARN

Scikit-learn is a powerful Python library for machine learning, offering tools for tasks like classification, regression, and clustering. It provides efficient implementations of popular algorithms, including support vector machines, decision trees, and k-means clustering. With its user-friendly API and compatibility with libraries like NumPy and Pandas, it is widely used for both research and practical applications.

4.3.4 XG BOOST

XGBoost (Extreme Gradient Boosting) is a powerful and efficient machine learning library designed for gradient boosting algorithms. It excels in handling structured/tabular data for tasks like regression, classification, and ranking, offering high performance and scalability. With features like regularization, parallel processing, and handling missing data, XGBoost is widely used in competitive machine learning and real-world applications**.**

4.3.5 JOBLIB

Joblib is a Python library designed for efficient serialization and parallel computing. It is commonly used to save and load machine learning models or large data structures, offering faster performance compared to standard Python serialization methods like pickle. Joblib also supports parallel processing, making it useful for optimizing computationally intensive tasks.

4.3.6 SEABORN

Seaborn is used in crop yield prediction for visualizing data during exploratory data analysis. It helps identify relationships between variables like rainfall and yield, analyze distributions, examine correlations, and understand the impact of categorical factors. These visual insights assist in feature selection and improving model accuracy.

5.DESIGN REQUIREMENT ENGINEERING

CONCEPT OF UML:

UML (Unified Modeling Language) diagrams serve as a visual tool to represent the structure and behavior of a system in a clear and organized manner. These diagrams help to depict various aspects of the system, such as its components, roles, interactions, and operations, by using standardized symbols and notations. The primary purpose of UML diagrams is to improve the understanding of the system's architecture and design, making it easier to communicate complex ideas among developers, stakeholders, and team members. They also play a crucial role in system documentation, providing a blueprint that can be used for system development, modification, or maintenance. By creating a visual representation of the system, UML diagrams facilitate better decision-making, enhance collaboration, and ensure that the system's design meets the required specifications and user needs.

UML DIAGRAMS:

The Unified Modeling Language (UML) is a standardized language used for modeling the design and structure of systems across a variety of domains, including software engineering, business processes, and hardware design. Its primary purpose is to provide a visual representation of a system's architecture, similar to how blueprints guide the construction of buildings in engineering. UML helps teams organize complex information by breaking down the system into manageable visual components, which is particularly valuable in large-scale projects. With its set of standardized notations and symbols, UML allows for a clear depiction of both the static structure and dynamic behaviors of a system, such as the relationships between objects, user interactions, and system processes.

In complex applications with multiple teams involved, clear and effective communication is critical, especially when interacting with stakeholders who may not have a deep understanding of the underlying code. UML bridges this gap by offering a way to represent the system's requirements, features, and processes visually, making it easier for non-technical stakeholders to grasp the design and functionality of the system. By illustrating key processes, user interactions, and system structure, UML promotes collaboration among development teams, enhances understanding, and streamlines the development process. This visual approach not only improves communication but also ensures that everyone involved has a unified understanding of the system, ultimately contributing to more efficient and successful project execution.

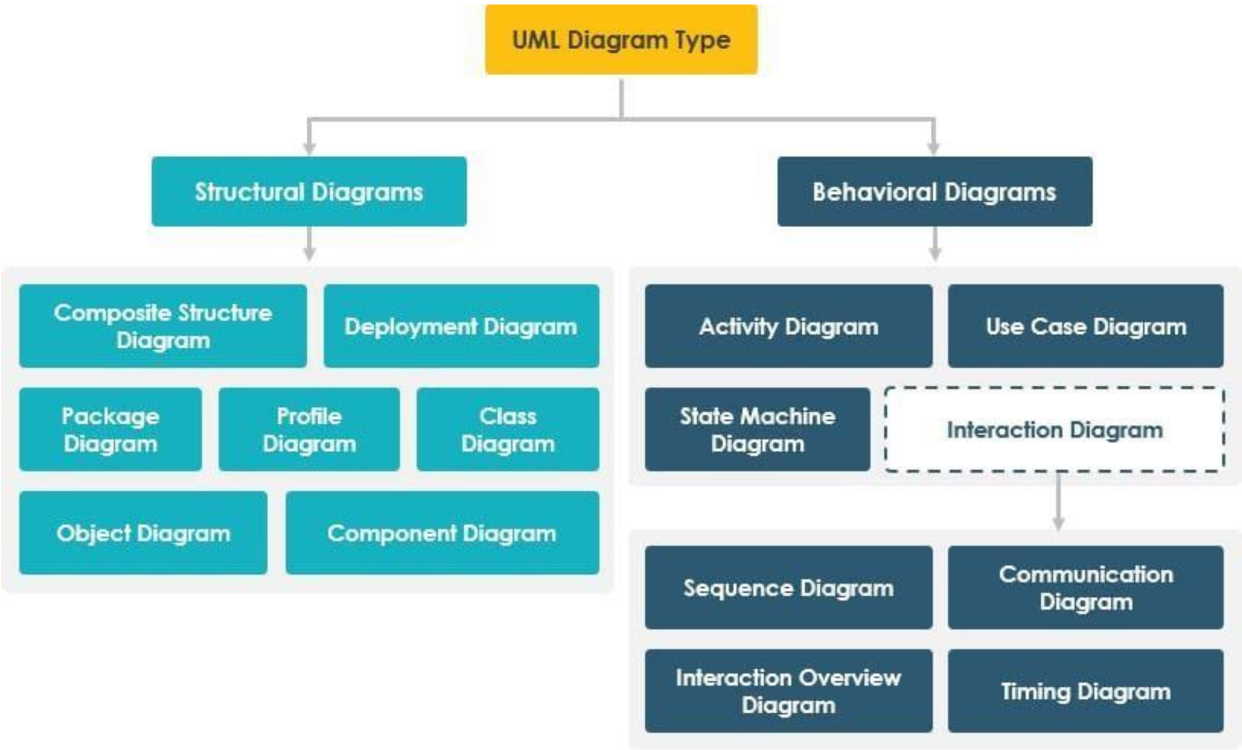


Figure:5 Concepts of uml

5.1 Use case Diagram:

The Use Case diagram for the Medical Prescription Recognition system illustrates the interactions between doctors, pharmacists, and the system. It highlights key functionalities such as uploading prescriptions, preprocessing images, recognizing and validating text, storing data, and retrieving and generating reports from the database.

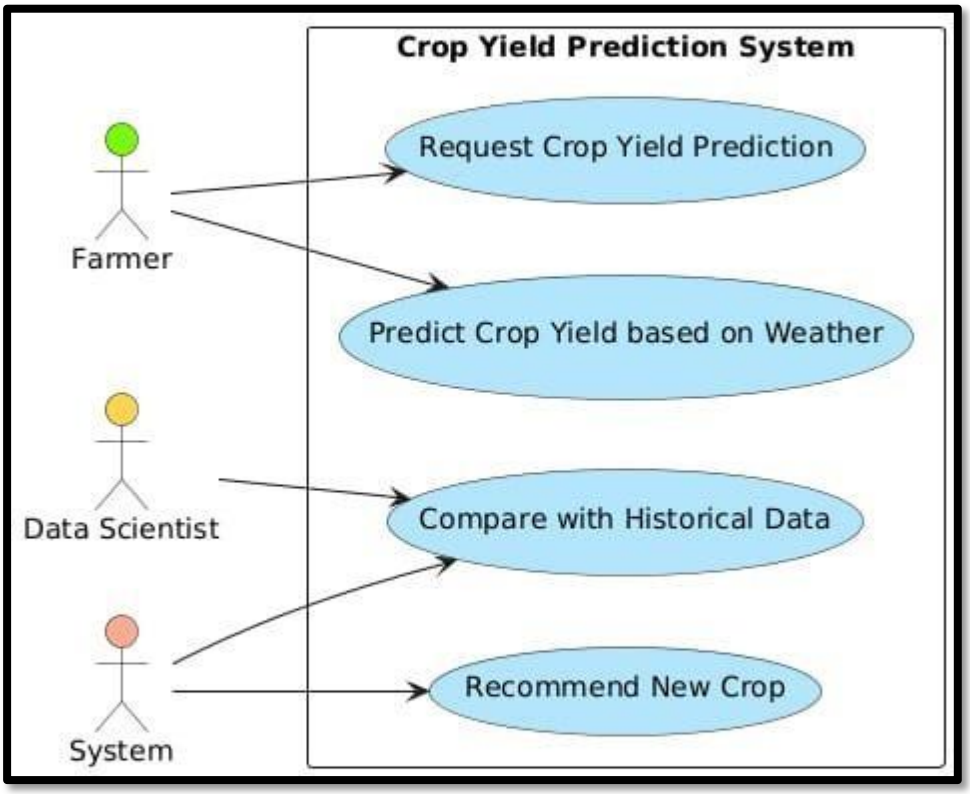


Figure:6 UseCase Diagram

5.2 Class Diagram

A class diagram is a static type of structural diagram that visually represents the architecture of a system by illustrating the connections among the system's classes, attributes, operations, and relationships. It provides a blueprint of how different components of the system interact and collaborate, showcasing the structure in a clear and organized manner.

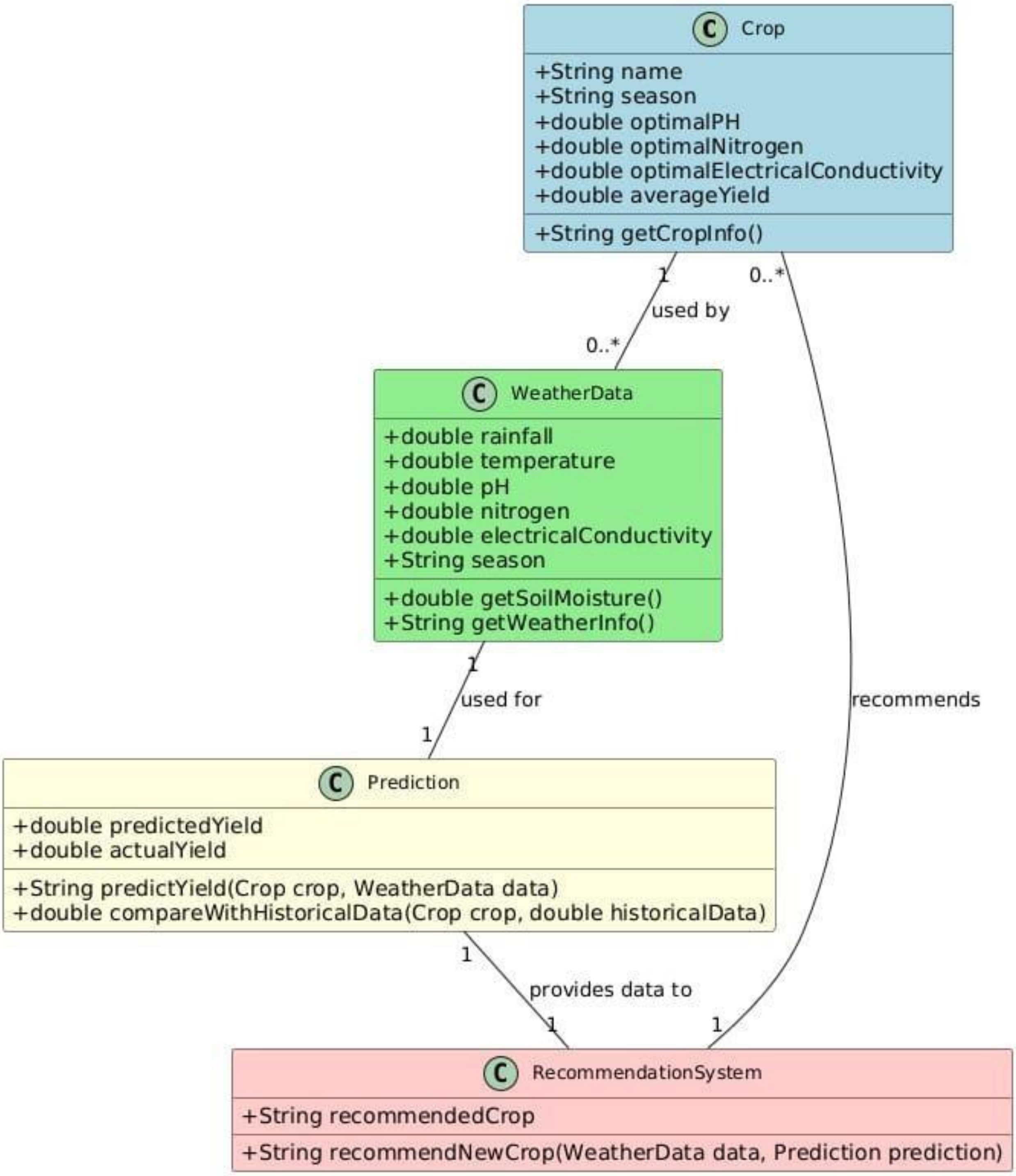


Figure:7 Class Diagram

5.3 Activity diagram:

An activity diagram in UML illustrates the sequence of actions and decisions within a system or process. It shows how activities interact and flow from start to finish, making it easy to understand and analyze complex workflows and business processes.

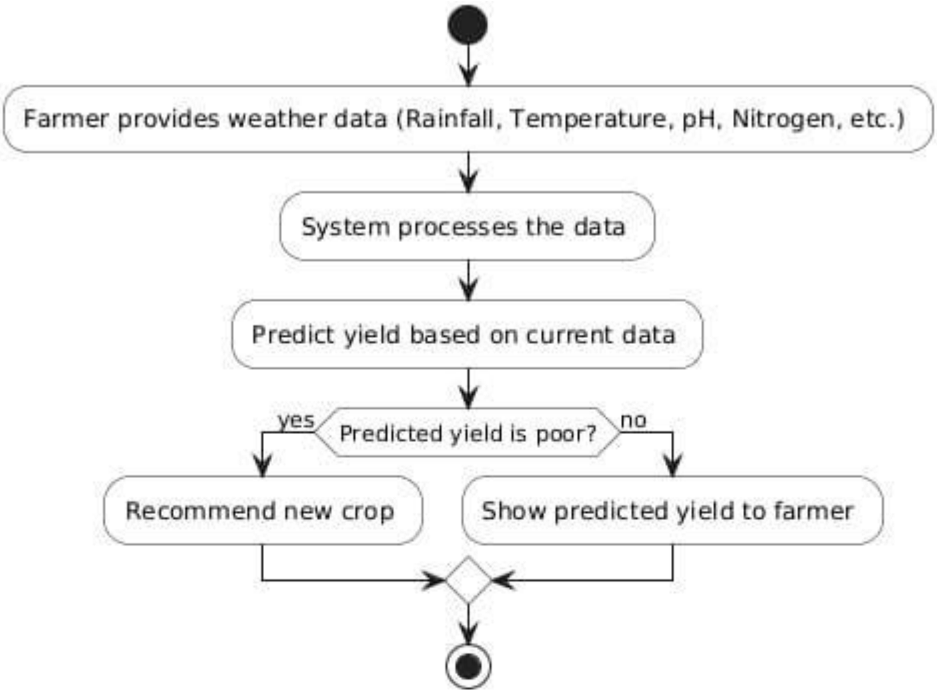


Figure:8 Activity Diagram

5.4 Sequence Diagram

A sequence diagram illustrates interactions between objects in a sequence over time, showing the messages exchanged between them. It visually represents the flow of control and data between objects in a system, emphasizing the order of events and the lifeline of each participating object.

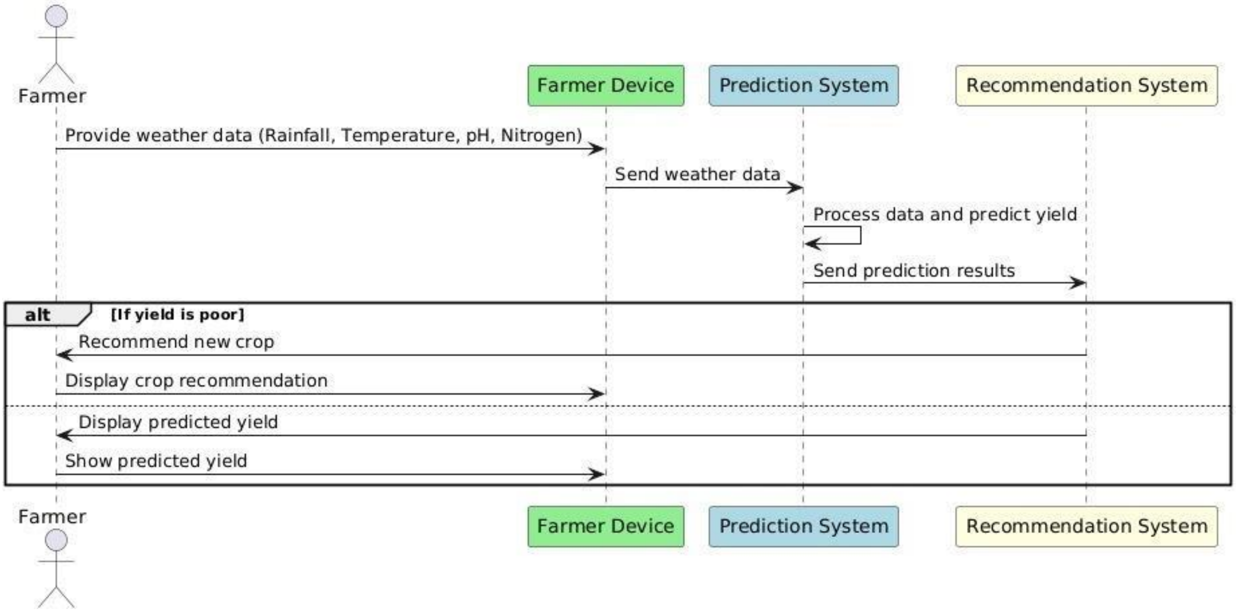


Figure:9 Sequence Diagram

5.5 System Architecture

1. **Data Sources**: The system collects data from weather (temperature, humidity), chemical (soil nutrients), pesticides (types and quantities), and yield (historical crop data) sources.
2. **ELT Process**: Data is extracted, loaded into a central database, and transformed through cleaning, handling missing values, and standardizing formats.
3. **Centralized Dataset**: The cleaned data is consolidated into a single dataset for analysis and modeling.
4. **Statistical Analysis**: Statistical methods identify key factors influencing crop yields, helping prioritize important features for modeling.
5. **Feature Engineering**: Relevant features like soil pH and rainfall are extracted, and unnecessary features are removed to improve model efficiency.
6. **Machine Learning**: The Random Forest algorithm trains the model on the dataset to learn relationships between input features and crop yields, followed by model validation.
7. **Yield Prediction**: The trained model predicts crop yields based on real-time conditions and recommends suitable crops for specific scenarios.

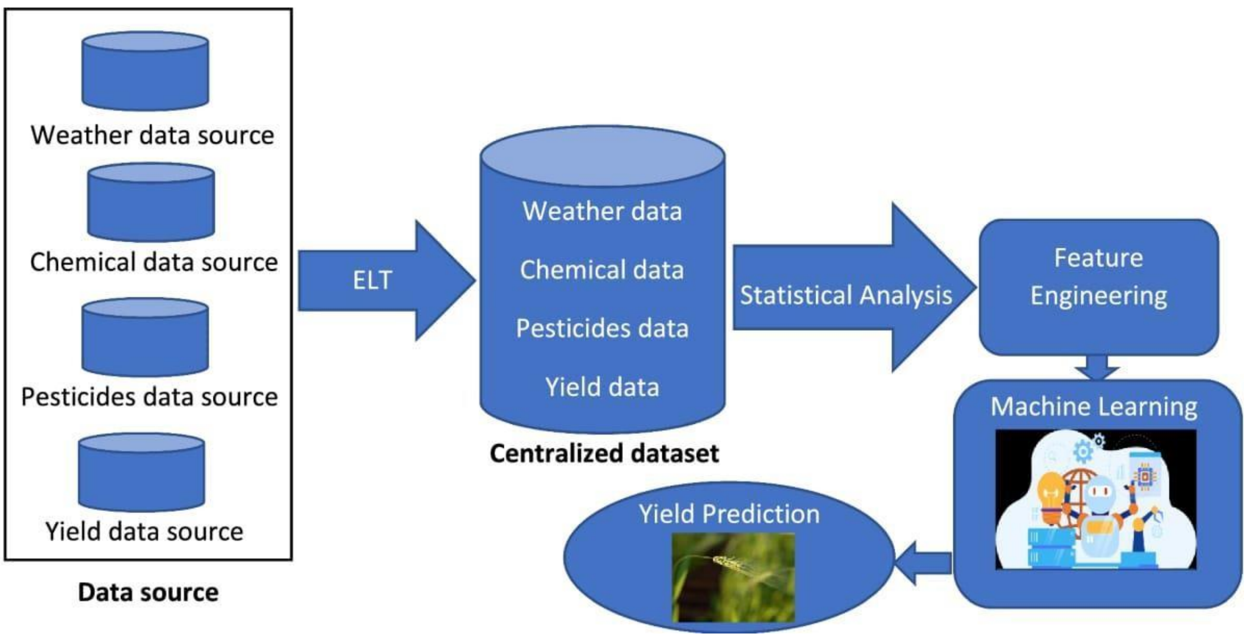


Figure:10 System Architecture

6. IMPLEMENTATION

The implementation part is done in Jupyter notebook. The data set contains the attributes like temperature, area, season, rainfall, nitrogen etc.

Dataset: https://www.kaggle.com/opencvmlpython/crop-yield-dataset2017?select=NewCropTrainFinal.csv

Area	Production	Rainfall	Season	Temperature	Nitrogen (kg/ha)	Electrical Conductivity (ds/m)
7800	3200	30.4	Kharif	28.0	497	4.1
39922	75572	111.9	Kharif	27.2	473	3.9
44656	49099	3.4	Rabi	20.3	366	4.9
6540	3945	30.9	Rabi	24.2	417	3.8
2911	2062	189.2	Kharif	27.5	267	3.5
7	2	21.6	Rabi	23.6	435	3.4
18	21	206.3	Summer	28.8	370	2.0
39157	1878596	186.9	Kharif	27.9	312	5.3
10562	16812	81.6	Kharif	26.2	334	3.8
179	2174	68.5	Rabi	24.1	365	2.9

Figure:11 Sample Dataset

Dataset Statistics:

Total Entries: 85,256 records

Training Count: 59,679 records (used for model training)

Testing Count: 25,577 records (used for model testing)

Features:

Area: The area of the land in square meters (numeric).

Production: The yield or amount produced in kilograms (numeric).

Rainfall: The amount of rainfall in millimeters (numeric).

Season: The crop-growing season (categorical: Kharif, Rabi, Summer).

Temperature: The temperature in degrees Celsius (numeric).

Nitrogen (kg/ha): The amount of nitrogen applied in kilograms per hectare (numeric).

Electrical Conductivity (ds/m): The electrical conductivity of the soil (numeric).

Importing Necessary Libraries:

Random Forest

```
+ Code   + Markdown
```

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

%matplotlib inline
```

Python

Figure:12 Importing Necessary Libraries

Importing and Describing the Dataset:

Importing Data

```python
df = pd.read_csv(r'C:\Users\DELL\Desktop\MAJ\Agri-Management\Trainset.csv')
df
```

Python

```python
df.columns
```

Python

```
Index(['Area', 'Production', 'Rainfall', 'Season', 'Temperature', 'Crop', 'pH',
       'Nitrogen(kg/ha)', 'ElectricalConductivity(ds/m)'],
      dtype='object')
```

Empty markdown cell, double-click or press enter to edit.

```python
df['Crop'].nunique()
```

Python

```
15
```

```python
df['Season'].nunique()
```

Python

```
6
```

```python
df.describe()
```

Python

Figure:13 Importing and Describing the Dataset

Data Visualization:

Data visualization

```python
cultivation_data = df[['Season','Crop', 'Area','Rainfall', 'Production','Temperature','pH','Nitrogen(kg/ha)']]
y='Maize'
cultivation_data=cultivation_data.groupby('Crop').get_group(y)
cultivation_data=cultivation_data.groupby('Season')[['Production','Rainfall', 'Area','Temperature','pH','Nitrogen(kg/ha)']].su

print(cultivation_data)
print("Bar plot of the above data")
cultivation_data.dropna().plot(kind='bar', figsize=(20,10), logy=True, color=['dodgerblue', 'orange','red','green','yellow','b
print("Line plot of the above data")
cultivation_data.dropna().plot(figsize=(20,10), logy=True, color=['dodgerblue', 'orange','red','green','yellow','black'], line
```

Python

```
              Production      Rainfall        Area     Temperature        pH  \
Season
```

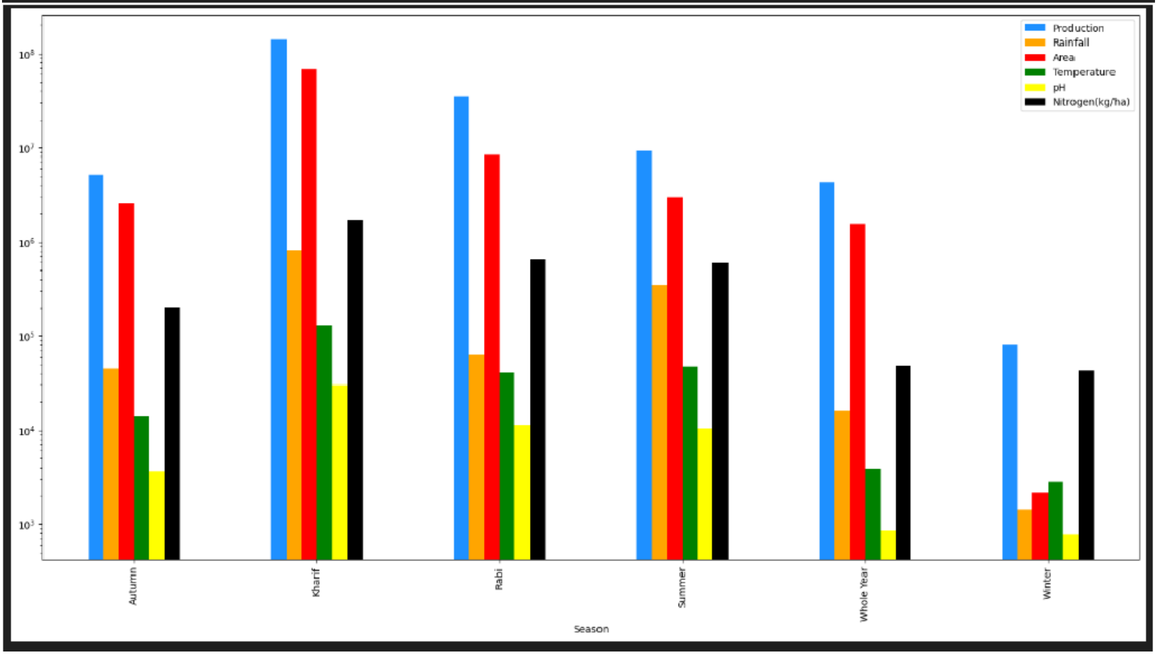
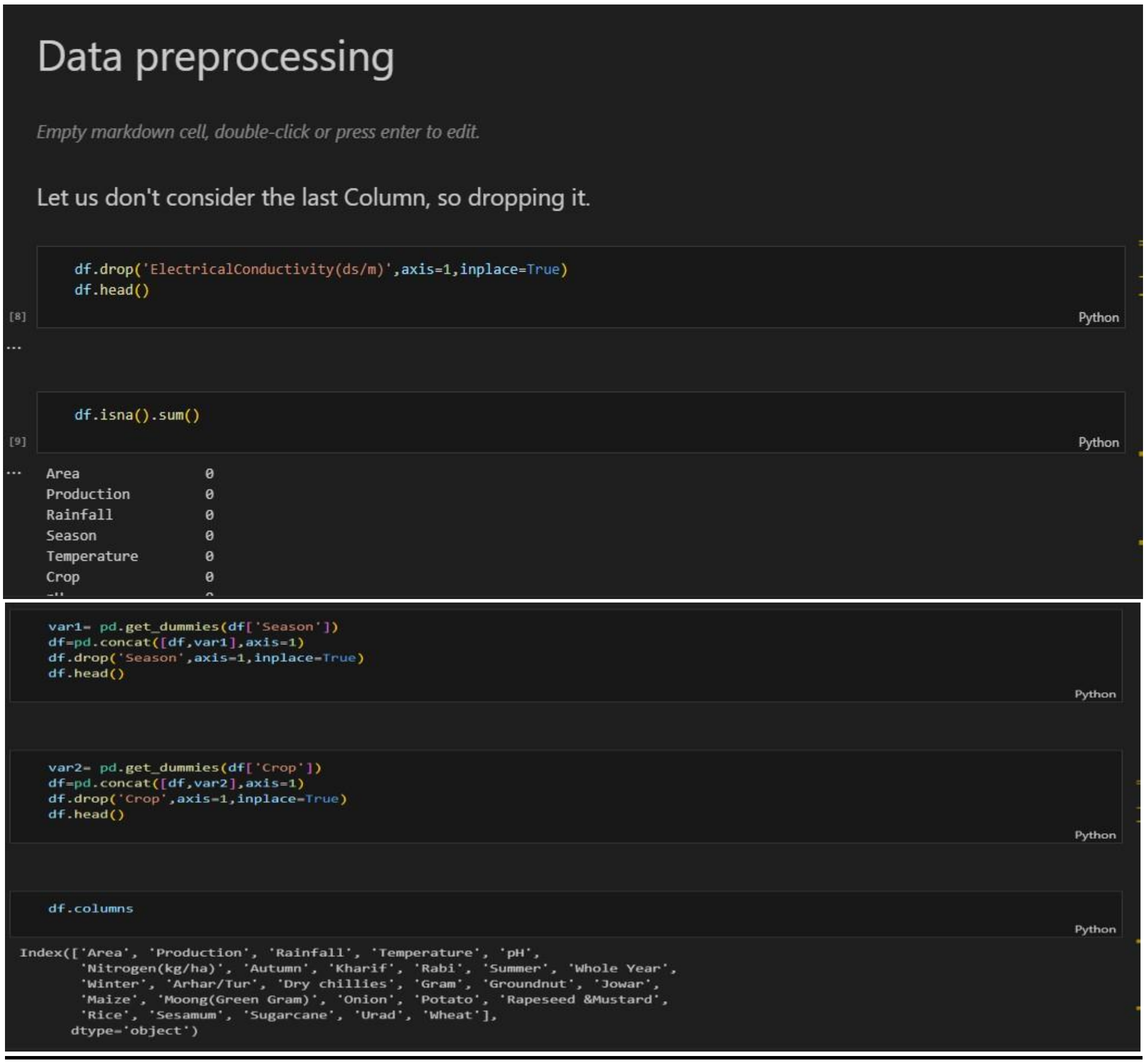


Figure:14 Data Visualization

Data Preprocessing:



Data preprocessing

Empty markdown cell, double-click or press enter to edit.

Let us don't consider the last Column, so dropping it.

```python
df.drop('ElectricalConductivity(ds/m)',axis=1,inplace=True)
df.head()
```

[8] Python

...

```python
df.isna().sum()
```

[9] Python

```
Area           0
Production     0
Rainfall       0
Season         0
Temperature    0
Crop           0
```

```python
var1= pd.get_dummies(df['Season'])
df=pd.concat([df,var1],axis=1)
df.drop('Season',axis=1,inplace=True)
df.head()
```

Python

```python
var2= pd.get_dummies(df['Crop'])
df=pd.concat([df,var2],axis=1)
df.drop('Crop',axis=1,inplace=True)
df.head()
```

Python

```python
df.columns
```

Python

```
Index(['Area', 'Production', 'Rainfall', 'Temperature', 'pH',
       'Nitrogen(kg/ha)', 'Autumn', 'Kharif', 'Rabi', 'Summer', 'Whole Year',
       'Winter', 'Arhar/Tur', 'Dry chillies', 'Gram', 'Groundnut', 'Jowar',
       'Maize', 'Moong(Green Gram)', 'Onion', 'Potato', 'Rapeseed &Mustard',
       'Rice', 'Sesamum', 'Sugarcane', 'Urad', 'Wheat'],
      dtype='object')
```

Figure:15 Data Preprocessing

Defining Random Forest Model:

```python
from sklearn.ensemble import RandomForestRegressor
```
Python

```python
from sklearn.model_selection import train_test_split

train_inputs, val_inputs, train_targets, val_targets = train_test_split(
    X, y, test_size=0.2, random_state=0)
```
Python

```python
rfc = RandomForestRegressor(random_state=0,n_jobs=-1)
```
Python

```python
rfc.fit(train_inputs,train_targets)
```
Python

```python
rfc.score(train_inputs,train_targets)
```
Python

```
0.9698391615854721
```

```python
rfc.score(train_inputs,train_targets)
```
Python

```
0.9698391615854721
```

```python
rfc.score(val_inputs,val_targets)
```
Python

```
0.78317047915743
```

```python
pred = rfc.predict(val_inputs)
```
Python

```python
from sklearn.metrics import mean_squared_error
from math import sqrt
```
Python

```python
rms = sqrt(mean_squared_error(val_targets,pred))
rms
```
Python

```
305653.310757631
```

Figure:16 Defining Random Forest Model

Hyperparameter Tuning Using GridSearchCV for Random Forest:

```
from sklearn.model_selection import GridSearchCV
```

Python

```
n_estimators = [int(t) for t in np.linspace(start=10,stop = 100, num=10)]

max_features = ['auto', 'sqrt']

max_depth = [5,10,15,20]

min_samples_split = [2,4,6]

min_samples_leaf = [1,2,3]

bootstrap = [True,False]
```

Python

```
param_grid = {
                'n_estimators' : n_estimators,
                'max_features' : max_features,
                'max_depth' : max_depth,
                'min_samples_split' : min_samples_split,
                'min_samples_leaf' : min_samples_leaf,
                'bootstrap' : bootstrap
                }
```

Python

```
rf_Grid = GridSearchCV(estimator = rfc, param_grid = param_grid, cv = 2, n_jobs=4)
```

Python

```
rf_Grid.fit(train_inputs,train_targets)
```

Python

```
rf_Grid.best_params_
```

Python

```
{'n_estimators': 30}
```

```
param_grid = {

                'max_features' : max_features
}
```

Python

```
rf_Grid.best_params_
```

Python

```
{'n_estimators': 30}
```

```
rf = RandomForestRegressor(random_state=0,n_jobs=-1,
                n_estimators = 30,
                max_depth = 15,
                min_samples_split = 6,
                bootstrap= True)
```

Python

Figure:17 Hyperparameter Tuning Using GridSearchCV

Evaluation Metrics for Random Forest Model:

```
[51]
rf.score(train_inputs,train_targets)

0.9392156400829137

[52]
rf.score(val_inputs,val_targets)

0.7891670825267723
```

Figure:18 Evaluation Metrics for Random Forest Model

Defining Decision Tree Model:

```python
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt

# Load the dataset
df = pd.read_csv(r'C:\Users\DELL\Desktop\MAJ\Agri-Management\Trainset.csv')

# Data preprocessing (One-hot encoding for categorical variables)
df.drop('ElectricalConductivity(ds/m)', axis=1, inplace=True)
var1 = pd.get_dummies(df['Season'])
df = pd.concat([df, var1], axis=1)
df.drop('Season', axis=1, inplace=True)

var2 = pd.get_dummies(df['Crop'])
df = pd.concat([df, var2], axis=1)
df.drop('Crop', axis=1, inplace=True)

# Normalize continuous features
df['Area'] = df['Area'] / df['Area'].max()
df['Rainfall'] = df['Rainfall'] / df['Rainfall'].max()
df['Temperature'] = df['Temperature'] / df['Temperature'].max()
df['pH'] = df['pH'] / df['pH'].max()
df['Nitrogen(kg/ha)'] = df['Nitrogen(kg/ha)'] / df['Nitrogen(kg/ha)'].max()
```

```python
# Define features (X) and target variable (y)
X = df.loc[:, df.columns != 'Production']
y = df['Production']

# Split data into training and validation sets
train_inputs, val_inputs, train_targets, val_targets = train_test_split(X, y, test_size=0.2, random_state=0)

# Initialize Decision Tree Regressor
dtr = DecisionTreeRegressor(random_state=0)

# Train the Decision Tree model
dtr.fit(train_inputs, train_targets)

# Predictions for Decision Tree
pred_dtr = dtr.predict(val_inputs)
```

Figure:19 Defining Decision Tree Model

Evaluation Metrics for Decision Tree Model:

```python
# Accuracy and RMSE for Decision Tree
dtr_train_accuracy = dtr.score(train_inputs, train_targets)
dtr_val_accuracy = dtr.score(val_inputs, val_targets)
rms_dtr = sqrt(mean_squared_error(val_targets, pred_dtr))
va_dtr = rms_dtr / df['Production'].max()

# Print results for Decision Tree
print(f"Decision Tree Training Accuracy (R²): {dtr_train_accuracy * 100:.2f}%")
print(f"Decision Tree Validation Accuracy (R²): {dtr_val_accuracy * 100:.2f}%")
print(f"Decision Tree RMSE: {rms_dtr}")
print(f"Decision Tree RMSE Normalized: {va_dtr}")
```

```
Decision Tree Training Accuracy (R²): 100.00%
Decision Tree Validation Accuracy (R²): 73.61%
Decision Tree RMSE: 337181.15536966233
Decision Tree RMSE Normalized: 0.011905273475378234
```

Figure:20 Evaluation Metrics for Decision Tree Model

Defining XgBoost Model:

```python
import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt

# Load the dataset
df = pd.read_csv(r'C:\Users\DELL\Desktop\MAJ\Agri-Management\Trainset.csv')

# Data preprocessing (One-hot encoding for categorical variables)
df.drop('ElectricalConductivity(ds/m)', axis=1, inplace=True)
var1 = pd.get_dummies(df['Season'])
df = pd.concat([df, var1], axis=1)
df.drop('Season', axis=1, inplace=True)

var2 = pd.get_dummies(df['Crop'])
df = pd.concat([df, var2], axis=1)
df.drop('Crop', axis=1, inplace=True)

# Normalize continuous features
df['Area'] = df['Area'] / df['Area'].max()
df['Rainfall'] = df['Rainfall'] / df['Rainfall'].max()
df['Temperature'] = df['Temperature'] / df['Temperature'].max()
df['pH'] = df['pH'] / df['pH'].max()
df['Nitrogen(kg/ha)'] = df['Nitrogen(kg/ha)'] / df['Nitrogen(kg/ha)'].max()
```

```python
# Define features (X) and target variable (y)
X = df.loc[:, df.columns != 'Production']
y = df['Production']

# Split data into training and validation sets
train_inputs, val_inputs, train_targets, val_targets = train_test_split(X, y, test_size=0.2, random_state=0)

# Initialize XGBoost Regressor
xgb_model = xgb.XGBRegressor(objective='reg:squarederror', random_state=0, n_jobs=-1)

# Train the XGBoost model
xgb_model.fit(train_inputs, train_targets)

# Predictions for XGBoost
pred_xgb = xgb_model.predict(val_inputs)
```

Figure:21 Defining XgBoost Model

Evaluation Metrics for XGBoost Model:

```python
# Accuracy and RMSE for XGBoost
xgb_train_accuracy = xgb_model.score(train_inputs, train_targets)
xgb_val_accuracy = xgb_model.score(val_inputs, val_targets)
rms_xgb = sqrt(mean_squared_error(val_targets, pred_xgb))
va_xgb = rms_xgb / df['Production'].max()

# Print results for XGBoost
print(f"XGBoost Training Accuracy (R²): {xgb_train_accuracy * 100:.2f}%")
print(f"XGBoost Validation Accuracy (R²): {xgb_val_accuracy * 100:.2f}%")
print(f"XGBoost RMSE: {rms_xgb}")
print(f"XGBoost RMSE Normalized: {va_xgb}")
```

```
XGBoost Training Accuracy (R²): 99.33%
XGBoost Validation Accuracy (R²): 78.86%
XGBoost RMSE: 301771.3991243338
XGBoost RMSE Normalized: 0.01065501727011983
```

Figure:22 Evaluation Metrics for XGBoost Model

7. SOFTWARE TESTING

Software testing is the process of testing before the real software is run through to the end. Ensuring that the expected output is free from mistakes and faults is the primary goal of software testing.

7.1 Unit Testing:

Unit testing is the first step in validating individual components within the Agriculture Yield Prediction System to ensure that each function or module performs as expected in isolation. The preprocessing module was rigorously tested to handle missing values, normalize data, and detect and remove outliers effectively, ensuring clean and standardized data input for predictions. The feature extraction process was validated to confirm that relevant features, such as soil moisture, temperature, and rainfall, were accurately identified and fed into the machine learning model. Core machine learning algorithms, including decision trees, random forests, and linear regression, were individually tested to verify their training process, accuracy, and capability to predict crop yields accurately based on input data. Additionally, model evaluation metrics such as accuracy, precision, recall, and F1 score were applied to assess the performance of the algorithms, ensuring the system delivers reliable predictions.

7.2 Integration Testing:

Integration testing plays a crucial role in validating that the components of the Agriculture Yield Prediction System work seamlessly when combined. This testing phase ensures that the interactions between various modules of the system function correctly and produce consistent results. Key focus areas for integration testing included the connections between data preprocessing, the machine learning model, and the user interface.

The Data Preprocessing to Machine Learning Model integration was tested to ensure that the system effectively handled the flow of cleaned and preprocessed data into the machine learning algorithm. This involved verifying that the input data retained its integrity and that transformations, such as normalization and feature selection, were correctly implemented before being passed to the model. Errors such as data mismatches, missing features, or incorrect formats were identified and resolved during this phase.

For the Model to Output Display, the predictions generated by the machine learning model were tested to ensure they were correctly formatted and transmitted to the output display. This step validated the system's ability to translate the technical model output into a user-friendly format, enabling end-users, such as farmers, to easily interpret the results and make informed decisions. Attention was given to the clarity, accuracy, and responsiveness of the output display, ensuring real-time insights without lag or errors.

The System Interactions testing verified the overall communication between all modules. It ensured that the system handled dependencies, data flows, and error handling robustly. Scenarios such as interrupted data flows, unexpected inputs, or system failures were simulated to check how well the modules interacted and recovered from errors. This stage confirmed the alignment of inputs, processes, and outputs across the system, ensuring consistent and reliable operation in real-world scenarios.

Furthermore, integration testing also considered performance aspects, such as system response time during interactions and the ability to handle concurrent requests from multiple users. This phase of testing was essential to build a cohesive, scalable, and robust system capable of delivering accurate and actionable insights in agricultural environments.

7.3 Acceptance Testing:

Acceptance testing was conducted to ensure that the Agriculture Yield Prediction System met all business and user requirements, focusing on its practicality and effectiveness from an end-user perspective. The user interface was reviewed by farmers and stakeholders to confirm that it was user-friendly, intuitive, and easy to navigate. The system was evaluated for responsiveness, ease of use, and overall usability to ensure it provided a seamless experience. Functionality validation involved testing the core features, such as data input, yield prediction, and result interpretation, to ensure they aligned with user needs. Farmers confirmed that the predictions were accurate, relevant, and useful for making informed agricultural decisions. Additionally, usability testing included gathering feedback from stakeholders to address issues and incorporate suggestions for improvement. This involved refining how predictions were displayed and ensuring the system could handle diverse input scenarios effectively. Performance testing ensured the system could process real-time data inputs and deliver predictions quickly, enabling timely decision-making, which is critical in agricultural applications. This comprehensive approach to acceptance testing ensured the system’s readiness for practical deployment.

7.4 Testing on our System:

The Agriculture Yield Prediction System leverages machine learning algorithms to forecast crop yields based on factors such as soil quality, rainfall, temperature, and historical data, providing farmers with data-driven insights to optimize crop production, mitigate risks, and enhance agricultural efficiency. A thorough testing strategy was implemented to ensure the system's reliability, involving unit testing, integration testing, acceptance testing, and system testing.

During unit testing, individual modules like data preprocessing, feature extraction, machine learning algorithms, and model evaluation were validated in isolation to ensure accurate and efficient functioning. Integration testing focused on verifying the seamless interaction between components, such as the flow from data preprocessing to machine learning models and the presentation of predictions on the user interface, ensuring consistency and error-free communication across modules. Acceptance testing validated the system’s usability and functionality from the perspective of farmers and stakeholders. The interface was tested for user-friendliness, and feedback was incorporated to refine features like prediction display and handling diverse input scenarios. Performance testing confirmed that the system could process real-time inputs and deliver quick results for time-sensitive decision-making.

Finally, **system testing** evaluated the entire system under real-world conditions, testing its scalability, stress-handling capability, security, and cross-platform compatibility. By subjecting the system to diverse datasets, extreme conditions, and multi-platform environments, its robustness and reliability were ensured. This comprehensive testing approach guarantees that the system meets functional, performance, and user-experience standards, making it a reliable tool for aiding farmers in making informed agricultural decisions. With its accurate predictions, intuitive interface, and robust design, the system stands ready to improve crop yields and transform agricultural practices.

8. RESULTS

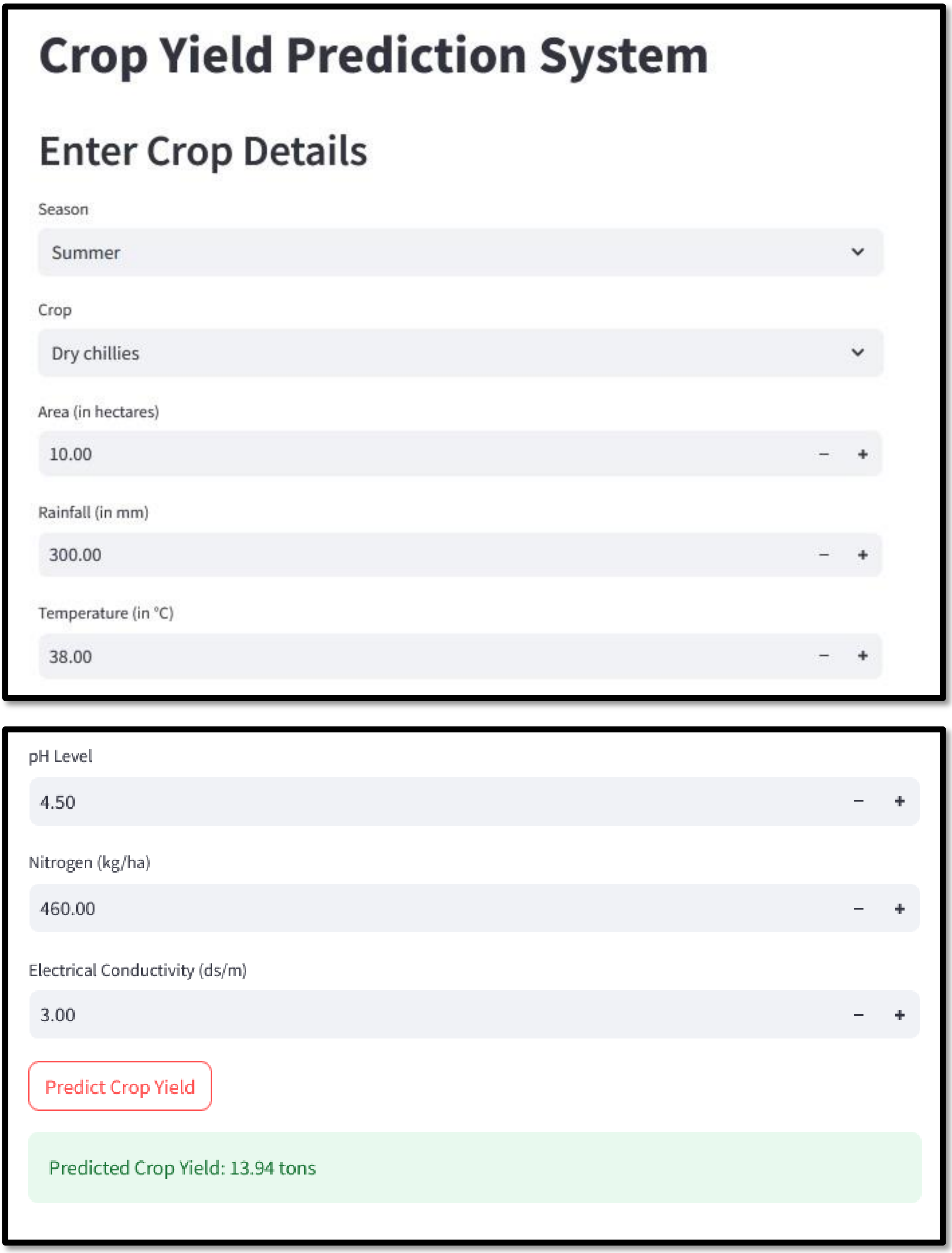


Figure:23 Final Output

9. CONCLUSION AND FUTURE ENHANCEMENTS

The conclusion of this project highlights the significant impact of machine learning techniques, particularly the Random Forest regressor, in improving crop yield prediction. By utilizing historical data, such as weather patterns, soil conditions, and previous crop yields, the model is able to accurately forecast crop productivity. This predictive capability provides farmers with valuable insights into the potential success of their crops, enabling them to make more informed decisions about which crops to cultivate based on the expected yield. Ultimately, this enhances agricultural efficiency and productivity, benefiting not only farmers but also contributing positively to the overall economy, especially in a country like India, where agriculture plays a pivotal role in sustaining the economy.

Furthermore, the project integrates a crop recommendation system that uses past data to suggest the most suitable crops for cultivation in specific regions. This system provides a tailored approach to farming, considering the unique environmental and soil conditions of different areas. By recommending the most appropriate crops based on data-driven insights, farmers can optimize their cultivation practices, reducing risk and increasing the chances of a successful harvest. The ability to choose the right crop, in turn, maximizes yield rates, contributing to better food production and stability for the country’s agricultural sector.

Finally, the system’s potential for future enhancement is an exciting prospect. One key area for improvement lies in integrating additional data sources, such as real-time soil fertility information, to further optimize recommendations. The inclusion of advanced techniques like IoT devices for real-time monitoring and more granular environmental data could lead to even more accurate predictions and actionable recommendations for farmers. Expanding the system’s reach to cover all of India could provide a national-scale solution for optimizing crop production, increasing food security, and boosting agricultural exports. By continuously improving the system and incorporating new data, this project can make a long-lasting, scalable impact on India’s agricultural practices, ultimately contributing to the nation’s economic growth.

10. BIBLIOGRAPHY

[1] https://ieeexplore.ieee.org/abstract/document/9432236/

[2] https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1128388/full?utm_source

[3] https://arxiv.org/abs/2308.08948?utm_source

[4] https://www.mdpi.com/2073-4395/14/10/2264?utm_source

[5] https://arxiv.org/abs/2208.12633?utm_source

[6] https://pmc.ncbi.nlm.nih.gov/articles/PMC8211294/?utm_source