MULTI-CRITERIA DECISION ANALYSIS APPROACH FOR CROP YIELD PREDICTION

A PROJECT REPORT

Submitted by

MOHAMED IMAD MASOOD T A (210071601097)

MOHAMED HASIN F (210071601125)

Under the guidance of

Dr. D. MADHINA BANU

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APRIL 2025

B.S. Abdur Rahman®
Crescent
Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956

BONAFIDE CERTIFICATE

Certified that this project report "**MULTI CRITERIA DECISION ANALYSIS APPROACH FOR CROP YIELD PREDICTON**" is the bonafide work of "**MOHAMED IMAD MASOOD T A (210071601097)** and **MOHAMED HASIN F (210071601125)**" who carried out the project work under my supervision. Certified further, that to the best of our knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. D. MADHINA BANU
SUPERVISOR
Associate Professor
Department of CSE
B S Abdur Rahman Crescent
Institute of Science and Technology
Vandalur, Chennai – 600 048

SIGNATURE

Dr. AISHA BANU W
HEAD OF THE DEPARTMENT
Professor
Department of CSE
B S Abdur Rahman Crescent
Institute of Science and Technology
Vandalur, Chennai – 600 048

B.S. Abdur Rahman®
Crescent
Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956

VIVA VOCE EXAMINATION

The viva voce examination of the **CSD 4201 - Project Work** titled **"MULTI CRITERIA DECISION ANALYSIS APPROACH FOR CROP YIELD PREDICTON",** submitted by **MOHAMED IMAD MASOOD T A (210071601097) and MOHAMED HASIN F (210071601125) ,** is held on_____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We sincerely express our heartfelt gratitude to Prof. **Dr. T. MURUGESAN**, Vice Chancellor and **Dr. N. THAJUDDIN**, Pro-Vice Chancellor, B.S. Abdur Rahman Crescent Institute of Science and Technology, for providing us an environment to carry out our course successfully.

We sincerely thank **Dr. N. RAJA HUSSAIN**, Registrar for furnishing every essential facility for doing our project.

We thank **Dr. SHARMILA SANKAR**, Dean, School of Computer, Information and Mathematical Sciences for her motivation and support.

We thank **Dr. W. AISHA BANU,** Professor and Head, Department of Computer Science and Engineering, for providing strong oversight of vision, strategic direction, and valuable suggestions.

We express our sincere thanks to the Project Review Committee members, **Dr. S.P.VALLI**, Associate Professor, and **Dr. D. MADHINA BANU,** Associate Professor, Department of Computer Science and Engineering for their valuable suggestions and support.

We obliged our project supervisor **Dr. D. MADHINA BANU,** Associate Professor, Department of Computer Science and Engineering for her professional guidance and continued assistance during our project.

We thank our class advisor, **Dr. C. VIJAYALAKSHMI,** Assistant Professor (Senior Grade), Department of Computer Science and Engineering for her guidance and encouragement throughout the project period.

We thank all the **Faculty Members** and the **System Staff** of the Department of Computer Science and Engineering for their valuable support and assistance at various stages of project development.

MOHAMED IMAD MASOOD T A

MOHAMED HASIN F

ABSTRACT

Agriculture is a crucial aspect of maintaining world food security and economic stability but, facing modern-day challenges like unpredictable climatic conditions, soil erosion, and ineffective agriculture practices, crop yield prediction has become more complex. This work outlines an end-to-end machine learning-based system for crop type classification and yield estimation from real-world agricultural data with more than 100,000 records. The system uses significant environmental and soil factors such as nitrogen, phosphorus, potassium, temperature, humidity, rainfall, pH, state, district, and season. The most appropriate crop is predicted using a RF Classifier with a precision of 98%, while yield estimation is carried out using a RF Regressor with an R^2 value of 0.91 and mean absolute error (MAE) of 112.3 kg/ha. The model was then trained following thorough preprocessing, feature importance analysis, and low-impact feature removal to improve fairness and minimize overfitting—specifically, removing biases such as those experienced by Arecanut. The Streamlit-based user interface enables farmers to enter values, get crop recommendations sorted by confidence score, and see corresponding yield predictions, with input validation and a yield suppression mechanism for crops with confidence less than 1.0% included. Future developments involve integration of real-time weather APIs, IoT-based soil sensing, market price prediction via LSTM or ARIMA models, fertilizer suggestion systems, crop rotation scheduling, intelligent irrigation recommendations, and a multilingual AI-based chatbot for voice-guided support. This smart, scalable solution fills the gap between conventional agriculture and data-driven decision-making, enabling sustainable and precision agriculture practices.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION		FULL FORM
AI	-	Artificial Intelligence
API	-	Application Programming Interface
ARIMA	-	Auto Regressive Integrated Moving Average
CSV	-	Comma-Separated Values
GIS	-	Geographic Information Systems
IoT	-	Internet of Things
LSTM	-	Long Short-Term Memory
MAE	-	Mean Absolute Error
ML	-	Machine Learning
MSE	-	Mean Squared Error
PCA	-	Principle Component Analysis
R^2	-	Coefficient of Determination
RF	-	Random Forest
RFE	-	Recursive Feature Elimination
RMSE	-	Root Mean Squared Error
ROC	-	Receiver Operating Characteristic
SVM	-	Support Vector Machine
WSN	-	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

Agriculture is extremely important in the world economy as it plays a pivotal role in food production, providing raw materials, and offering employment to millions of individuals. Though it's very crucial, crop yield predictions are still the biggest challenge faced because of many environmental factors that affect it such as soil productivity, changing patterns of weather, and the ever-increasing influence of climate change. Traditional prediction methods, based heavily on past experience and expert judgment, are frequently inadequate in today's fast-evolving agricultural environment.

Thanks to the advent of data science and machine learning (ML), new possibilities have emerged to enhance both the precision and effectiveness of yield predictions. This proposed work will develop a machine learning-based crop yield prediction model from important agricultural indicators like nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, and rainfall. The ultimate goal is to create a smart, adaptive system that can assist farmers in making better decisions regarding crop selection and projected yield, ultimately increasing productivity with reduced losses.

1.1 OVERVIEW

In the age of contemporary agriculture, crop yield prediction has emerged as a crucial factor in planning and decision-making. It enables farmers to make appropriate crop choices, utilize inputs well, and time harvesting more effectively. Nevertheless, the conventional overdependence on history and experiential wisdom commonly proves unable to offer reliable forecasts in the face of unstable environmental fluctuations like climatic changes or soil erosion.

This proposed work overcomes those constraints by utilizing machine learning methods that examine critical agricultural variables to improve prediction accuracy. Both classification and regression models are used—a combination of RF for identifying appropriate crops and regression models for predicting yield levels.

By providing the model with information on soil nutrients, weather data, and seasonal variables, it provides data-driven insights to inform farmers' decisions.

Along with fundamental prediction features, the system takes into account possible upgrades such as adding real-time weather forecasts and IoT sensors for observing soil conditions in order to further improve its usability.

This technology-based approach is consistent with sustainable agriculture objectives because it minimizes resource waste, decreases uncertainty, and increases effective food production.

1.2 DESCRIPTION

Crop yield prediction is one of the most important determinants of agricultural success. It aids in optimizing production, farm input management, and planning for future requirements. Traditional prediction techniques are, however, usually inadequate because they fail to account for variability in environmental conditions such as unanticipated rainfall, temperature fluctuations, and soil composition alterations.

This proposed work proposes a machine learning solution that utilizes key agricultural parameters—i.e., N, P, K, temperature, humidity, and rainfall—to predict crop yields. The model incorporates classification models to identify the most appropriate crops and regression methods to calculate yield output. A complete pipeline including data preprocessing, feature engineering, model training, and testing guarantees solid and scalable predictions.

The platform is not just for enhancing the quality of farming decisions but also to help ensure sustainable agriculture by limiting losses and resource wastage. It further creates opportunities for future growth with functionalities such as real-time climate integration, IoT sensor-based inputs, and even market price forecasts to further enable agricultural stakeholders.

1.3 OBJECTIVES

The primary aim of this proposed work is to develop an intelligent, accurate, and efficient machine learning system for predicting crop yield, enabling farmers to make informed choices and maximize land productivity. The system intends to harness both environmental and soil-related data to deliver actionable insights.

- **Building a reliable prediction model:** Employ algorithms like RF and regression methods for crop classification and yield forecasting.
- **Supporting farmer decision-making:** Offer insights tailored to soil quality, weather, and seasonal data to guide crop selection.
- **Boosting prediction accuracy:** Focus on key inputs—N, P, K, temperature, humidity, and rainfall—to ensure high-quality predictions.
- **Using real-world agricultural data:** Train models on diverse datasets representing multiple regions to improve adaptability.
- **Exploring future potential:** Investigate integrations such as live weather feeds, IoT-based soil monitoring, and economic forecasting tools like crop market prices.

This proposed work contributes to the field of precision agriculture by merging traditional farming wisdom with intelligent, data-driven technologies.

1.4 ABOUT THE PROJECT

This proposed work focuses on the creation of a machine learning-based system aimed at predicting accurately both the best crops to plant and their corresponding yields. The system uses several environmental and soil parameters to make its predictions, providing a thorough, data-based decision support system for the purposes of contemporary agriculture. Its main objective is to assist farmers, agronomists, and agricultural policymakers in making effective, strategic choices that conform with both environmental conditions and resource availability.

Fundamentally, the model unites sophisticated machine learning algorithms to analyze key agronomic features including the amounts of vital soil nutrients such as nitrogen (N),

phosphorus (P), and potassium (K) and chief climate factors including temperature, humidity, and rain. The model also takes account of seasonal variation and geographical variations to make predictions that are context-specific and correct. These inputs are carefully preprocessed using steps such as data cleaning, normalization, and feature selection in order to achieve high-quality, informative datasets for training and evaluation. The machine learning pipeline follows a structured workflow.

- **Data preprocessing:** Raw agricultural data is cleaned to remove inconsistencies and formatted for analysis.
- **Feature selection:** The most relevant attributes influencing crop yield are identified to enhance model performance.
- **Model training:** Various machine learning algorithms are trained using historical data to learn patterns and relationships.
- **Prediction and evaluation:** The trained models are validated using real-world data to assess their accuracy and reliability in yield estimation and crop classification.

CHAPTER 2

LITERATURE SURVEY

Shreya S. Bhanose and Kalyani A. Bogawar [1], carried out a study called "Crop And Yield Prediction Model" presented in the International Journal of Advance Scientific Research and Engineering Trends, suggested a two-stage machine learning method that could help farmers make data-driven decisions regarding crop selection and yield prediction. The model employed historical agricultural information including variables like temperature, precipitation, soil type, and crop yield to initially suggest the ideal crop with a Naïve Bayes classifier and then estimate the predicted yield with linear regression using features like crop productivity, soil type, rainfall, and temperature. Highlighting interpretability and simplicity, the authors attempted to create a lean model that could be added to user-accessible systems for real-world usage. The research highlighted the need for effective preprocessing of data, such as normalization and missing value handling, to improve model performance. Although the model was reasonably accurate, constraints like inability to adapt in real time and use of simple algorithms were pointed out. The authors recommended that future research could include increasing the feature set, using dynamic inputs, and employing more sophisticated methods for improved scalability and prediction accuracy.

Tripathy, A. K. [2], carried out a study called "Data Mining and Wireless Sensor Network for Agriculture Pest/Disease Predictions" at World Congress on Information and Communication Technologies, investigated an integrated solution through the integration of data mining technologies with WSNs for predicting agricultural diseases and pests. The research suggested a smart monitoring system that could gather real-time environmental information—temperature, humidity, and soil moisture—via installed sensors, which was then processed with machine learning algorithms to identify early indicators of pest infestation or disease outbreaks. Utilizing decision tree-based classification techniques, the system sought to deliver timely and accurate alerts to farmers, enabling them to proactively manage crop health and reduce possible losses. The authors pointed out the benefit of employing WSNs for real-time, remote monitoring,

minimizing reliance on manual inspection and allowing more accurate data gathering. In spite of the encouraging findings, the study recognized network scalability, sensor node energy efficiency, and region-specific pest model requirements as challenges. In general, the paper showed the capability of integrating IoT-based sensing with smart data analysis to improve decision-making in precision agriculture and minimize crop susceptibility to pests and diseases.

Ramesh Babu Palepu [3] carried out a study called "An Analysis of Agriculture Soils by Using Data Mining Techniques" to investigate the possibility of data mining techniques to enhance soil management and increase agricultural productivity. The study stressed that soil quality is a fundamental component of effective crop farming, and using data-based methods can improve decision-making greatly. The research applied methods like k-means clustering and plant residue analysis through images to evaluate soil properties and detect trends that might inform subsequent farming practices. Such methods made it possible to predict likely events in soil performance and nutrient levels so that intervention could be proactive. In a follow-up study, V. Rajeswari et al. utilized classification algorithms such as JRip, J48, and I Bayes to forecast and classify various types of soil, and specifically Red and Black soil. Of these, the JRip algorithm produced the best results, as indicated by enhanced Kappa Statistics. Their research proved the effectiveness of decision trees and Bayesian networks in the investigation of soil fertility and aiding specific agricultural planning. In concert, these researches portrayed the useful application of data mining to precision agriculture in terms of understanding soil condition and contributing to sustainable agriculture practices as well as enhancing crop performance.

Rajeswari and K. Arunesh [4], presented the paper "Analysing Soil Data using Data Mining Classification Techniques," presented in the Indian Journal of Science and Technology, investigated the usage of different data mining classification algorithms in analyzing soil data for better agricultural decision-making. The main aim of the study was to categorize soil types and evaluate their potential for various crops depending on qualities like pH, electrical conductivity, organic carbon content, and macronutrient status like nitrogen, phosphorus, and potassium. The research utilized various machine learning

methods like Decision Trees, Naïve Bayes, and Support Vector Machines (SVM) to compare their performance in predicting correctly the soil profiles. Among these, Decision Tree algorithms proved good interpretability and fair accuracy and were a practical option for actual implementation. The authors stressed the need for accurate soil classification to direct crop choice, fertilizer application, and land management measures. Though the findings were encouraging, the research cited limitations in the form of the quality and quantity of the dataset and added that the use of additional real-time and geospatial data could improve model accuracy. On the whole, the study identified how data mining methodologies can be successfully employed to convert raw soil data into decision-making information for sustainable agriculture.

A. Swarupa Rani [5], presented the paper "The Impact of Data Analytics in Crop Management based on Weather Conditions" published in the International Journal of Engineering Technology Science and Research, discussed how data analytics can contribute to crop management practices substantially by using weather-based information. The research emphasized studying the impact of major meteorological parameters like temperature, humidity, and rainfall on crop growth cycles and productivity. Through the application of predictive analytics and statistical models, the study was intended to predict ideal sowing periods, watering schedules, and areas of risk owing to weather fluctuations. The research brought to the forefront the importance of big data and sophisticated computational software in recognizing trends and patterns between weather patterns and crop yield, thereby allowing more informed farmer decision-making. The paper also examined the need for combining historical weather data with real-time information in order to develop adaptive crop management strategies that are capable of responding to dynamic climatic conditions. While the results highlighted the advantages of data-driven agriculture, challenges such as data quality, multi-source integration, and the requirement for user-friendly tools were also noted. In general, the study emphasized the transformative potential of weather-based data analytics in enhancing efficiency, minimizing losses, and developing sustainable agricultural practices.

Pritam Bose [6] contributed a novel research work entitled "Spiking Neural Networks for Crop Yield Estimation Based on Spatiotemporal Analysis of Image

Time Series", proposing the application of Spiking Neural Networks in remote sensing-based crop yield estimation. The authors created the first SNN-based computational model using Normalized Difference Vegetation Index image time series to estimate crop yields. Their model was used to estimate winter wheat yield in Shandong Province, China, with a high accuracy of 95.64%, an average prediction error of 0.236 t/ha, and a correlation coefficient of 0.801. The research proved the ability of SNNs to handle complex spatiotemporal data, showing their potential for large-scale agricultural monitoring. Moreover, the authors suggested the creation of a NeuCube-based system for real-world commercial use, placing their research at the cutting edge of intelligent precision agriculture technologies.

Priyanka P. Chandak [7] gave a paper called "Smart Farming System Using Data Mining", which suggested an intelligent agricultural system that uses data mining technologies to automatically improve crop yields and utilization of resources. The system is integrated with various sources of data such as satellite images, web-based weather reports, and soil test results maintained in central databases. By the use of clustering algorithms, the model provides precise tracking and analysis of different agricultural parameters like crop growth stages, climatic changes, water usage, fertilizer distribution, and pest control measures. The research brings out the fact that agriculture consumes almost 70% of the world's water, which makes optimized water management crucial in fulfilling increasing food demand. By aggregating valuable insights from big data, the smart farm system enables real-time tracking and enables data-based decision-making. Furthermore, since the system is autonomous, timely alerts for critical weather conditions are enabled, improving productivity, reducing wastage of resources, and promoting sustainable farming.

Vikas Kumar [8] offered a paper entitled "KrishiMantra: Agricultural Recommendation System", which suggested a semantic web-based architecture for providing personalized farm recommendations using spatial information and expert knowledge bases. The system utilizes Information and Communication Technology (ICT) for filling the gap between experts and farmers, thus improving decision-making based on geographic and climatic data. KrishiMantra integrates Geographic Information

Systems (GIS) and Semantic Web technology to process and provide answers to queries submitted by farmers using mobile phones. The queries are associated with GIS data and crop-specific knowledge bases to create location-aware and context-aware recommendations. The system processes important climate parameters like temperature, rain, humidity, and soil moisture to facilitate precision farming activities. The research further details future prospects for enhancing ontology-based data representation and user interface design towards ease of integration and increased accessibility by end-users within agricultural communities.

Savae Latu [9] offered a paper entitled "Sustainable Development: The Role of GIS and Visualization", discussing the use of Geographic Information Systems (GIS) and visualization technologies to ensure sustainable management of resources in agriculture. The study underscores the essential need to reconcile economic development with the conservation of the environment, especially for developing countries like Pacific island nations, where the two objectives frequently clash. GIS technology is showcased as a potent tool for visualization of intricate agriculture, environmental, and spatial data, thus enabling land-use planning, resource allocation, and ecosystem conservation in a more informed manner. Through facilitating decision-making using data, the research establishes the manner in which GIS may have an instrumental function in the attainment of sustainable agricultural development objectives as well as managing difficulties encountered by areas with low resources and high environmental sensitivity.

Nasrin Fathima G. [10], presented the paper titled "Agriculture Crop Pattern Using Data Mining Techniques" published in the International Journal of Advanced Research in Computer Science and Software Engineering, discussed applying data mining techniques to study and determine the best cropping patterns from historical agricultural data. The research sought to reveal significant patterns and interrelations among many factors like soil type, climatic conditions, crop type, and seasonal trends for facilitating more effective crop planning and decision-making. Methods such as classification, clustering, and association rule mining were used to handle large amounts of data to predict appropriate crop combinations for particular areas. The study highlighted the potential of such methods to assist farmers in making informed decisions regarding crop

rotation and land use, thereby ensuring enhanced yield and sustainability. The paper further discussed the significance of proper and recent data collection for enhancing the validity of the forecasts. Even with the challenges of data sparsity and regional variability, the research established that data mining has tremendous potential in the optimization of agricultural practices and in creating intelligent decision support systems for more effective crop management.

Ramesh A. Medar [11], presented study "A Survey on Data Mining Techniques for Crop Yield Prediction," presented in the International Journal of Advance Research in Computer Science and Management Studies, explored the use of various data mining techniques for predicting crop yield. The study reviewed multiple machine learning approaches, such as decision trees, regression analysis, and neural networks, to evaluate their effectiveness in crop yield prediction. The author emphasized the importance of historical data, including environmental factors like temperature, soil composition, and rainfall, in building accurate predictive models. Additionally, the paper discussed the impact of data preprocessing methods, such as handling missing values and data normalization, on model performance. Despite showcasing the effectiveness of these techniques in providing reliable yield predictions, the study highlighted challenges such as data quality, overfitting, and model generalization in real-world scenarios. The author suggested that future research should focus on improving model scalability, integrating real-time data inputs, and exploring hybrid models for enhanced prediction accuracy and adaptability.

CHAPTER 3

SYSTEM REQUIREMENTS AND DESIGN

3.1 PROBLEM DEFINITION

Agriculture is still the backbone of most national economies, a major source of livelihood and food supply. But even with advancements in farming implements and techniques, most farmers still grapple with selecting the most appropriate crops and estimating crop yields correctly. The reasons for this are numerous, unpredictable factors including changing weather patterns, soil nutrient loss, infestation by pests, and poor access to up-to-date and region-specific information on agriculture.

In the past, farmers have used experience, intuition, and general weather patterns to make their planting decisions. While this knowledge has been in use for generations, it lacks the accuracy necessary to handle the complexities that have been brought by the modern dynamics of farming. Therefore, such traditional methods might result in bad crop selection, wasteful utilization of resources, and finally, decreased productivity and financial loss.

As digital technologies are advancing at a lightning speed, machine learning has also turned out to be a useful tool to aid precision agriculture. By processing large amounts of data consisting of soil nutrient content, temperature, humidity, rainfall patterns, and past history of yields, machine learning algorithms can identify pattern correlations and provide highly accurate recommendations for crop choice and yield estimation.

The goal of this study is to create a predictive system based on machine learning methodologies that will assist farmers in making decisions. The system will suggest appropriate crops and yield estimates by utilizing historical information, existing environmental conditions, and soil report analysis. This solution not only enhances farming efficiency but also aids sustainable farming, reduces wastage of resources, and helps ensure economic stability for farmers. Additionally, this tool can be deployed through web and mobile platforms, making data-driven insights easily accessible even in remote agricultural regions.

Through connecting ancient agriculture to current analytics, this proposed work aims to empower farmers with the weapons they need to maximize yields and be able to deal with new challenges in the field.

3.2 EXISTING SYSTEM

In the majority of agricultural environments, especially rural and semi-urban, farmers still use traditional approaches to choosing crops and forecasting yields. These traditional methods are most often based on generational agricultural experience, informal knowledge sharing across the community, advice from farm extension officers, and information gathered from government literature. Although such sources give an initial understanding, they tend to be based considerably on generalized historical information and large-scale climatic patterns, more than specific, farm-level advice.

Conventional yield estimation is typically established on rudimentary practices like minimal soil testing employing hand kits, interpretation of climatic weather outlooks for each season, and an examination of past years' crop performances. These methods have the disadvantage that they are always limited in what they can forecast. They remain very sensitive to changes in conditions in the environment at the local level, with the potential of causing a major margin of error. Consequently, the process of decision-making becomes susceptible to uncertainty, which enhances the risks of underproduction, overproduction, and misallocation of resources.

Furthermore, such systems have no ability to track actual-time environmental changes and emerging agricultural problems. For example, sudden weather abnormalities—such as unseasonal rains, long dry seasons, or acute drops in temperatures—can greatly affect crop condition and productivity. Such dynamic changes cannot be handled and reacted upon by conventional systems, resulting in reactive instead of proactive agricultural administration.

Another drawback of current methods is their incapacity to accommodate large-scale, high-dimensional data. Modern agriculture necessitates the integration of multiple data sources including remote sensing imagery, IoT sensor readings, climate monitoring via satellites, soil moisture levels, and pest outbreak notifications. These older methods are

not capable of supporting such integration, which leaves them inadequate for precision farming applications today.

In addition, because of the lack of automation and predictive analytics, farmers are usually deprived of timely recommendations and insights that may assist them in optimizing inputs such as fertilizer, water, and pesticides. This leads to overuse or underuse of resources, directly influencing soil health and long-term sustainability.

In conclusion, though conventional crop selection and yield forecasting have been the pillars of agribusiness for decades, they are inadequate in the rapidly changing agro-environment of today. With the increasing sophistication of climate trends, market requirements, and environmental limitations, there is an urgent need for more dynamic, data-intensive, and intelligent systems that can provide real-time, local, and precise recommendations. Hence, the demand for contemporary AI and ML-driven solutions grows more and more paramount to facilitate the shift toward smart and sustainable farming.

3.2.1 Limitations of the Existing System

Although they have been in use for a long time, traditional practices of crop planning and yield calculation have a number of drawbacks:

- Traditional methods are not able to incorporate real-time information like soil nutrient deficiencies or abrupt weather conditions.
- Crop recommendations tend to be generic and are not specific to the needs of individual farms.
- Farmers spend time on cumbersome activities like farm visits, soil analysis, and consultations with experts.
- The absence of automation results in the process being slower and more susceptible to human error.
- Most farmers lack access to data analytics or weather forecasting technology.
- Valuable data, like historical crop yields or satellite images, are not maximally utilized.

- With shifting climatic patterns, conventional methods of prediction become inapplicable.
- Abrupt environmental changes tend to go unnoticed, leading to yield predictions that are inaccurate.
- Without accurate crop forecasting, there is a possibility of overutilization or underutilization of key inputs such as water, fertilizers, and pesticides.
- These inefficiencies can erode soil health and raise the cost of operations.

3.3 PROPOSED SYSTEM

The solution utilizes machine learning (ML) to address the internal constraints of traditional farming practices with the implementation of a data-driven, intelligent decision support system. This system is designed to help farmers make better and more informed decisions regarding crop choice and yield estimation through analysis of a mix of fundamental agronomic variables, including soil composition, temperature, humidity, rainfall patterns, pH levels, and past crop yield history.

In contrast to conventional techniques that are highly dependent on human judgment, generalized projections, or anecdotal experience, the ML-driven model is powered by algorithms that can identify intricate, non-linear patterns between many variables. The models are trained on varied and extensive databases, enabling them to pick out faint patterns and interplay** that may not be easily visible to the naked eye. In doing this, the system increases the accuracy and credibility of crop suggestions, providing farmers with a confidence score for every recommended crop and an estimated yield value.

One of the biggest advantages of the system is its personalization and flexibility. Instead of giving blanket suggestions, the solution offers personalized recommendations based on user-specific and real-time inputs. For instance, a farmer can feed in inputs such as the present nutrient levels of the soil, rainfall in the upcoming season, or recent temperature patterns, and the model will analyze this data to suggest the most appropriate crops for that place and period.

To make it accessible and easy to use, the model is integrated into a friendly interface, which can be accessed through both web and mobile applications. The system is designed in such a way that farmers with little technical knowledge can use the system easily. Users can enter parameters through drop-down menus, sliders, or plain text fields, and get simple, actionable outputs, such as

- The highest-ranked crops for their land,
- Projected yield suggestions for all crops, and
- Ancillary suggestions to enhance productivity.

Aside from crop scheduling, the system is also promoting effective utilization of resources. By reconciling crop selections to environmental patterns and resource inputs, it can allow farmers to avert excess uses of water, fertilizers, and pesticides, which lead to economic gain and ecologically friendly farm production as well. The environment is affected through less soil degradation, pollution, and carbon footprint.

In addition, the model is meant to improve over time. As additional data are gathered from various users and areas, the machine learning algorithms can be retrained and fine-tuned to become more predictive and responsive to new trends in agriculture, diseases, or climate variability.

Essentially, the solution fills the gap between ancient wisdom and contemporary science by providing a smart, scalable, and context-aware tool for farmers. It equips them with credible insights that minimize uncertainty, enhance productivity, and promote a transition to precision agriculture—the future of farming in an age characterized by climate change, food security threats, and digitalization.

3.3.1 Objectives of the Proposed System

The main goal of the suggested system is to create an intelligent crop advisor tool that recommends the most appropriate crops for planting based on soil nutrients such as nitrogen, phosphorus, and potassium, and weather conditions such as temperature, humidity, and rainfall. The recommendations are then narrowed down using location-based information such as the user's state and the season. To facilitate decision-making,

the system also generates confidence scores for each crop recommendation. Apart from recommendations, the tool is designed to give extremely accurate yield forecasts, providing predicted yields in kilograms per hectare by utilizing past data and crop performance trends. These forecasts also come with confidence scores to assist in avoiding spurious results, particularly for inappropriate crop selections. For ensuring fairness in prediction, the system descales biased features like Rainfall_Temperature, Weather_Index, and Humidity_Temperature that may bias outputs, thus ensuring balanced performance for all types of crops. A strong data preprocessing pipeline is deployed, using StandardScaler to scale numerical features and OneHotEncoder to transform categorical features like state and season. These operations are stored to apply them consistently during training and deployment stages. The system employs machine learning algorithms, such as RandomForestClassifier for classification and RandomForestRegressor for yield prediction, with optimal hyperparameter tuning to achieve maximum accuracy. An easy-to-use Streamlit interface is created to enable simple data entry by farmers, with interactive features and visual enhancements like crop images for improved usability. The application also incorporates input validation to check that input values fall within realistic ranges and provides helpful warnings when incorrect data is encountered, thus avoiding misleading outputs. To make the overall user experience better, the interface is styled with personalized color schemes, typography, and layout design, and results are displayed with visuals and background elements to make the application interactive. In addition, the tool includes location-based crop guidance to make the recommendations contextually appropriate and take into account regional and seasonal factors. Finally, to facilitate seamless real-time deployment, techniques such as @st.cache_resource in Streamlit are implemented to reduce processing time and prevent repetitive redundancy of preprocessing work upon repeated usage.

3.4 MODULE IDENTIFICATION

The system to be proposed is designed with a group of specialized modules, each having an indispensable function to facilitate effective, timely, and user-friendly crop suggestion and yield prediction. Central to this design is the Data Preprocessing Module, whose task

it is to manipulate raw agricultural data into clean, consistent, and machine-learning-compatible formats. This module addresses missing values in a systematic manner using statistical imputation methods—mean or median for continuous variables like Nitrogen (N), Phosphorus (P), Potassium (K), temperature, and humidity, and mode imputation for categorical fields like the state and season. To allow machine learning algorithms to learn from the data in the correct manner, categorical features are translated into numerical representations through OneHotEncoding with utmost care to maintain the identical structure of encoding during training and real-time prediction. In addition, this module identifies and removes features that can introduce bias into the model. Features like Rainfall_Temperature, Weather_Index, and Humidity_Temperature are dropped since they have the capacity to skew predictions, particularly for sensitive crops like Arecanut. Extra correlation analysis is done to identify and eliminate redundant or highly correlated features, hence enhancing model generalizability. Once cleaned, the dataset is divided into training and testing sets in an 80-20 proportion while maintaining class balance. All the numerical inputs are normalized using StandardScaler to standardize feature ranges and reduce differences due to unit variance, enhancing model convergence and performance.

Based on this, the Crop Prediction Module is responsible for identifying the most appropriate crops to be grown based on the input data provided by the user, such as soil nutrients, weather, and geographical location. It loads a pre-trained RandomForestClassifier model and converts user input into the same format as the training data to ensure consistency. The module produces a ranked list of crops with confidence scores that indicate each recommendation's certainty from the model. In order to preserve the reliability and quality of the recommendations, any prediction of a crop based on a confidence score below 1% automatically gets filtered out to not include crops upon which the model does not have confidence. The predictions are shown in an interactive visual format with high-resolution icons of crops and clever layout designs that can easily be interpreted by farmers.

Alongside this is the Crop Yield Estimation Module, aimed at predicting the predicted yield of every recommended crop in kilograms per hectare (kg/ha). This module is based on a

pre-trained RandomForestRegressor model, using the same preprocessed input features for the purpose of providing consistent and precise predictions. Notably, it only conducts yield estimation for those crops with a confidence score of 1% or more, thus saving on computational resources and minimizing the chance of producing untrustworthy results. Crops with a confidence score less than the threshold receive a default yield of 0 kg/ha, indicating that they are inappropriate for the current conditions. Yield values are merged with the crop recommendation output to provide a complete, integrated perspective for the user.

To ensure data integrity and avoid incorrect outputs, the Input Validation and Threshold Handling Module checks that all user-input values are within realistic agricultural limits. Specified operational ranges are nutrient values such as Nitrogen, Phosphorus, and Potassium between 10–120 units, temperature between 5°C and 50°C, and humidity between 10% and 100%. Where any input is outside the defined ranges, the system will immediately return a clear error message and stop execution to avoid making inaccurate or misleading predictions. This validation layer is important because it guarantees that the backend models only process reliable input data.

Further enhancing user interaction, the Streamlit User Interface Module provides a very presentable and highly interactive front-end interface. It includes interactive features like sliders for entering numeric values (e.g., soil nutrients and climate factors) and dropdown boxes for choosing categorical variables like state and season. This module enables users—especially farmers with little technical knowledge—to enter data in a straightforward manner and get real-time predictions as they adjust. To enhance user experience, the interface is enriched with customized color schemes, fashionable fonts, and large crop pictures. Background images and natural layouts are used to help users navigate through the process, and the application is not just informative but interactive as well. The UI incorporates real-time input validation at the interface level with instant feedback in case any input goes beyond valid limits, and the backend models are not executed in such scenarios.

Last but not least, the Data Storage and Model Caching Module is implemented to enhance application performance and resource usage. By using Streamlit's

@st.cache_resource decorator, the system stores both the classification and regression models in memory. This implies that the models do not have to be reloaded every time the user changes an input, which reduces prediction latency significantly. Temporary storage of inputs also enables users to adjust values without having to re-input previously entered data, making the experience smooth and seamless.

Collectively, the modules constitute a coherent and smart crop recommendation system. Every element functions together seamlessly to provide accuracy, ease, and effectiveness from raw data preprocessing to real-time user engagement, allowing the platform to be a dynamic tool for contemporary agriculture and farmer decision-making.

3.5 SYSTEM SPECIFICATION

3.5.1 Hardware Requirements:

Table 3.1 Hardware Requirements

Processor Type AMD RYZEN 7	**Speed** 4.40GHZ
RAM 16 GB RAM	**Hard disk** 1 TB
Keyboard 101/102 Standard Keys	**Mouse** Optical Mouse

3.5.2 Software Specification

Table 3.2 Software Requirements

Operating System Windows 10
Front End JUPYTER NOTEBOOK/ANACONDA TOOL
Coding Language PYTHON

3.6 DESIGN PROCESS AND EXPLANATION

The Crop Prediction and Yield Estimation System is built with a modular design that maximizes high accuracy, computational efficacy, and user-focused interaction. The process is divided into distinct stages—data preprocessing, model training, interface development, and deployment—and each performs independently but together optimized for best performance.The process begins with data acquisition and preprocessing. The database contains basic crop parameters such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, state, and season. For model enhancement, past data like yield history records and climate past are also added as external sources of data. Raw data is pre-processed where missing values are handled using imputation techniques, outliers eliminated, and categorical variables converted to numeric representations. Consistency is achieved in data by using methods of normalization like Min-Max scaling, whereby features are kept in comparable scales.

Feature selection is one of the most significant aspects of this phase. Through correlation analysis, features with least predictive performance contribution—Rainfall_Temperature, Weather_Index, and Humidity_Temperature—are discarded to prevent model biasing. Additionally, dimensionality reduction techniques such as PCA are employed in order to eliminate redundant variables and retain important data variance.

During the model building stage, the machine learning models are trained on two main tasks: crop classification and yield prediction. Classification models such as Logistic Regression, Decision Tree, RF, and XGBoost are trained to recommend suitable crops. Regression models such as Linear Regression and RF Regressor are employed for yield prediction. Hyperparameter tuning is employed to refine the accuracy of the models. Performance is evaluated based on appropriate measures such as accuracy, precision, recall, F1-score for classification problem, and MSE and R-squared for regression problems.

The user interface, made by Streamlit, is the interactive component of the system. It offers a way to input soil nutrient levels, climatic parameters, and region information to receive crop recommendations. Predictions are ranked in terms of model confidence. The interface features dynamic input validation in the event user input is beyond accepted ranges, then the system warns and prohibits incorrect predictions. Visual effects in the shape of customized themes, HD crop photos, and responsive structures improve the usability and the overall visual appeal.

3.6.1 Architecture Diagram

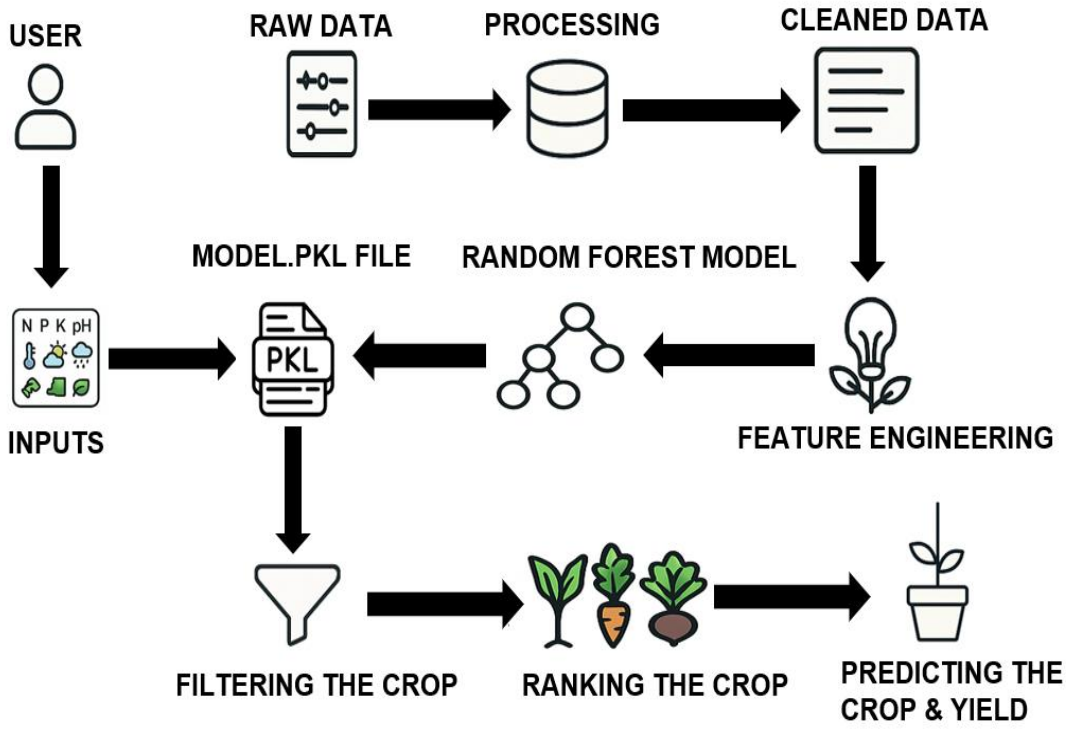

Figure 3.1 Architecture Diagram

The above Figure 3.1 depicts the overall architecture of the crop prediction system, starting with the gathering of raw farm data. The data is processed to remove inconsistencies and fill gaps to create a cleaned dataset. The cleaned data then goes through a feature engineering step where significant agronomic features like nitrogen (N), phosphorus (P), potassium (K), temperature, pH, and rainfall are harvested and cleaned. With this enriched data, a RF model is learned and stored as a .pkl for deployment. Upon receiving real-time input data from a user, the model applies this information to rank potential crops on the basis of suitability and ultimately predict the most suitable crop and its yield. This pipeline connects raw data to intelligent decision-making through machine learning for agricultural planning.

3.6.2 Flow Diagram

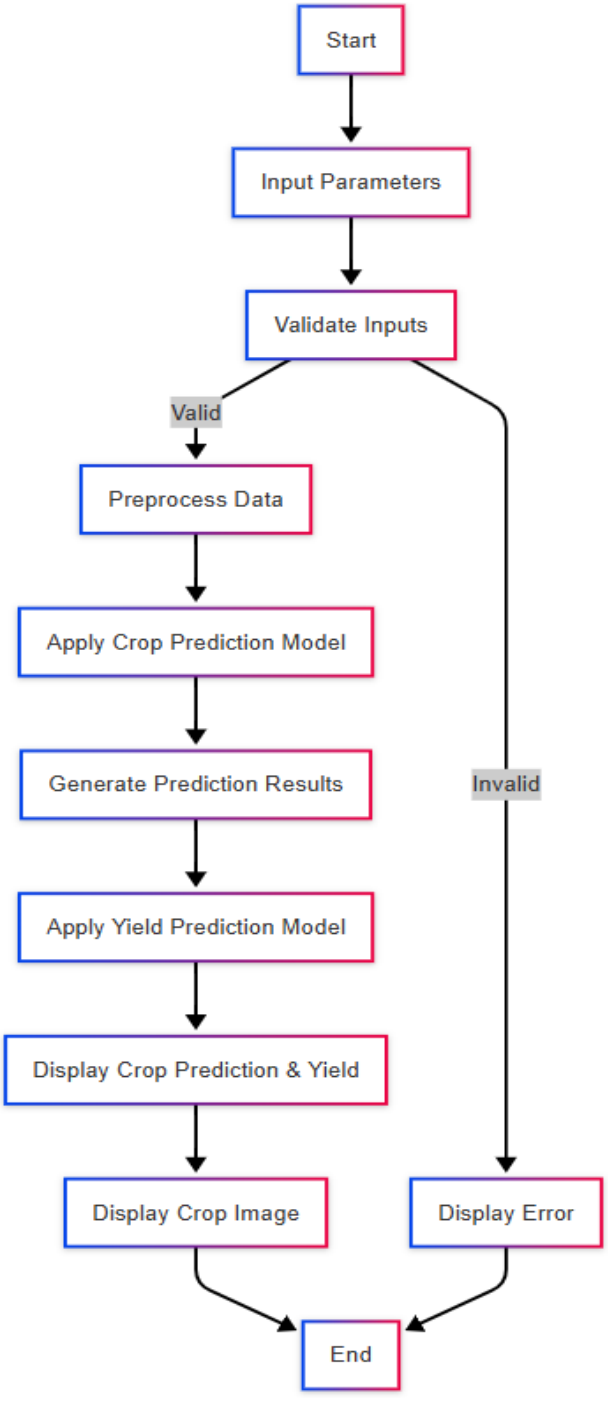

Figure 3.2 Flow Diagram

The above Figure 3.2 shows the crop and yield prediction system flow diagram, describing the step-by-step reasoning behind the application's functionality. The process starts with the user starting the system and providing input parameters like soil nutrients, temperature, pH, and rainfall. These inputs are initially validated—if the data is determined to be valid, it moves on to the preprocessing phase, where it is cleaned and formatted for model use. The preprocessed data is then input into the crop prediction model to produce preliminary results. These results are then passed through a yield prediction model to estimate possible output. The combined forecasts are then presented, both the suggested crop and its predicted yield. A matching crop image is also presented for user understanding. In case of failure of input validation at any stage, the system instantly returns an error message, providing strong handling of invalid data. This organized flow provides a seamless, consistent, and user-friendly forecasting experience.

CHAPTER 4

SYSTEM METHODOLOGIES

4.1 MODULE EXPLANATION

This section outlines the various modules of the crop prediction and yield estimation system. Each module focuses on a specific aspect of the machine learning pipeline, from data preprocessing to model evaluation.

4.1.1 Data Preprocessing Module

The Data Preprocessing Module encompasses all operations related to cleansing and structuring raw data prior to using it for model training. Major duties of this module are:

- Data Cleaning: This process is responsible for handling incomplete, redundant, or poorly formatted entries to make them consistent and accurate.
- Handling Categorical Data: Categorical inputs such as "State_Name" and "Season" are converted into numerical formats by applying encoding methods like One-Hot Encoding or Label Encoding.
- Feature Scaling: Numerical features, such as Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, and Rainfall, are normalized—usually by techniques such as StandardScaler—to provide equal weightage while training the model.
- Feature Engineering: New features can be developed to enhance the dataset. For example, creating a distinct "Crop_Code" to symbolize crop types numerically.

This module makes the dataset organized, trustworthy, and prepared for subsequent machine learning operations.

4.1.2 Feature Selection Module

The Feature Selection Module is concerned with determining the most influential features and removing irrelevant or redundant features. This assists in enhancing model performance as well as training efficiency.

- Irrelevant Feature Elimination: Features such as Rainfall_Temperature, Weather_Index, and Humidity_Temperature are removed because they might introduce unnecessary bias or noise.
- Correlation Analysis: Interrelationships between variables are examined in order to remove highly correlated features that could cause multicollinearity.
- Dimensionality Reduction: Methods like PCA or RFE are employed for reducing feature space without losing the important information.

The result is a clean dataset with only informative features that benefit model accuracy.

4.1.3 Model Training Module

This module is tasked with training machine learning models from the preprocessed dataset for both classification (crop type) and regression (prediction of the yield).

- Algorithm Selection: Ensemble methods such as RF are used for crop classification because they are strong and can deal with complex patterns. For yield estimation, a regression model like RF Regressor is used.
- Training Process: The sanitized dataset is divided into training and testing sets, and the model is trained on the training data.
- Hyperparameter Optimization: Model parameters are optimized with methods such as Grid Search or Random Search to improve performance.

Once trained, the model is serialized with tools like pickle for deployment in real-time prediction applications.

4.1.4 Prediction Module

This module connects the system and the user, producing real-time crop and yield predictions depending on the inputs given.

- User Input Collection: Users input critical inputs such as N, P, K, Temperature, Humidity, pH, Rainfall, State, and Season.

- Input Preprocessing: The module performs the same transformations performed while training (e.g., scaling and encoding) to make them compatible.
- Prediction Logic: The preprocessed input is fed through the trained crop classification and yield estimation models.
- Presentation of Output: Outputs are presented, indicating the most appropriate crops and estimated yield, as well as confidence scores and plots where relevant.

This module offers interpretable actionables that are useful for producing decisions.

4.1.5 Evaluation Module

The Evaluation Module checks the performance of the trained models and their reliability prior to deployment.

- Performance Metrics: For classification, Accuracy, Precision, Recall, and F1-score are computed. RMSE, MAE, and R^2 Score are used for regression models.
- Cross-Validation: Techniques of K-Fold cross-validation are applied to verify how accurately the model generalizes to unseen data.
- Model Benchmarking: Various models and parameter configurations are compared to choose the most effective configuration.
- Visual Interpretation: Results of evaluation are represented graphically using tools like Confusion Matrices, ROC Curves, and Residual Plots to gain insights into model behavior.

This module checks model effectiveness and makes sure that predictions are reliable and resilient under different conditions.

CHAPTER 5

IMPLEMENTATION

5.1 IMPLEMENTATION STEPS

A solid data preparation foundation is the starting point for the implementation of a trusted crop forecast and yield estimation system. This is achieved by collecting an ample dataset comprising essential agricultural parameters including nutrient content (N, P, K), climatic conditions (temperature, humidity, rainfall), soil pH, and geographical situation-specific parameters like state and season. After the data has been gathered, it is put through a detailed cleaning process to eliminate any inconsistency, missing value, or redundant records that would adversely affect model performance. In order to include more depth to the dataset, new features might be derived—such as seasonal average nutrients or regional soil quality ratings—which allow the model to better comprehend contextual relationships. In order to ready the data for machine learning algorithms, numerical features are normalized and categorical variables such as states and seasons are converted into a numerical form using methods like One-Hot Encoding or Label Encoding.

Once the dataset is prepared, the second area of focus is optimizing the features of the model. This entails examining how the various variables correlate with one another to identify and eliminate any that are too correlated, as they can cause bias and lower model performance. Non-essential or redundant attributes—like derived combinations such as Weather Index or Rainfall_Temperature—are also meticulously eliminated to simplify the model without sacrificing valuable information. If the data set is unusually large or complex, however, then dimensionality reduction techniques such as PCA may be used to reduce the data to its most informative elements, rendering the models more efficient and perhaps more accurate.

Once cleaned and optimized, however, then building and training the machine learning models is the next step. Two primary models are employed: a RF Classifier to forecast the best crop, and a RF Regressor to project the yield to be expected. These models are selected due to their consistency and robust performance on structured data with intricate relationships. The data is divided into training and test sets to enable the models to learn

and subsequently be tested fairly. Hyperparameters for every model are tuned using techniques such as Grid Search or Randomized Search to determine the optimal configuration. After training, the models are tested using standard measures—such as accuracy and F1-score for crop classification, and RMSE and R^2 score for yield regression—to verify that they are accurate and generalizable to new, unseen data.

After training and testing the models, they are incorporated into a simple-to-use web-based application. This interface, developed with Streamlit, enables users—farmers or agricultural experts—to enter values such as nutrient content, environmental information, and location. The system subsequently uses the same preprocessing that was performed while training the model to maintain consistency. Input validation checks within the built-in input assist users in entering values within practical agricultural limits, e.g., nitrogen levels of 10–120 or temperature values of 5–50°C, making the predictions reliable.

Having preprocessed the user inputs, the application makes predictions based on the trained models. The crop classifier generates a list of appropriate crops, along with confidence scores that indicate how likely a suggestion is. In order to prevent the presentation of unreliable choices, the system eliminates predictions with extremely low confidence. For the remaining crops, the model for yield estimation computes the predicted production per hectare. The outputs are provided in a straightforward and interactive way—usually featuring crop names, estimated yields, and supporting images—to enable the users to better visualize their choices and make better decisions.

The final stage is where the complete system is used as a responsive web application accessible via a web browser. Streamlit takes care of the front-end, with the backend encompassing all the trained models, data preprocessing pipelines, and the logic. To improve performance, caching mechanisms are employed to prevent unnecessary recomputation. Prior to going live, the application is thoroughly tested with a wide range of input scenarios to guarantee that it provides consistent, accurate results and stays stable even when presented with edge cases. This end-to-end technology not only gives precise predictions but also brings an easy-to-use interface that can handle real-life decision-making in agriculture so that farmers can enhance productivity and resource allocation based on science-based information.

5.2 MODULE-WISE SCREENSHOTS

The Data Preprocessing Module is a critical initial step in the machine learning pipeline, which guarantees that the raw data is cleaned, complete, and ready for proper model training and evaluation. The module is critical in ensuring data integrity and the quality of insights generated from the system.

Prior to using machine learning models, the dataset is checked for missing, inconsistent, and anomalous values. Missing values are a frequent problem in real-world agricultural datasets because of sensor malfunctions, human errors during manual entry, or incomplete history. Unhandled missing values can severely compromise model performance as well as introduce bias, so it is important that they be handled in preprocessing.

The procedure starts with a first scan of the dataset, whereby the system marks all columns that contain null or missing values. These are normally important agronomic variables like N (nitrogen), P (phosphorus), K (potassium), pH level, temperature, humidity, and rainfall. The count and percentage of missing entries per column are determined and presented so that the user can see how extensive the incompleteness of data is.

As shown in Figure 5.1, the module offers a side-by-side view of the dataset before and after preprocessing

The original dataset with missing values highlighted is shown in the left panel of the interface.

The right panel displays the cleaned dataset once the missing values have been filled in with appropriate methods like mean/mode/median imputation, KNN imputation, or regression-based filling, based on the type of variable and the data distribution.

5.2.1 Data Preprocessing Module Screenshot

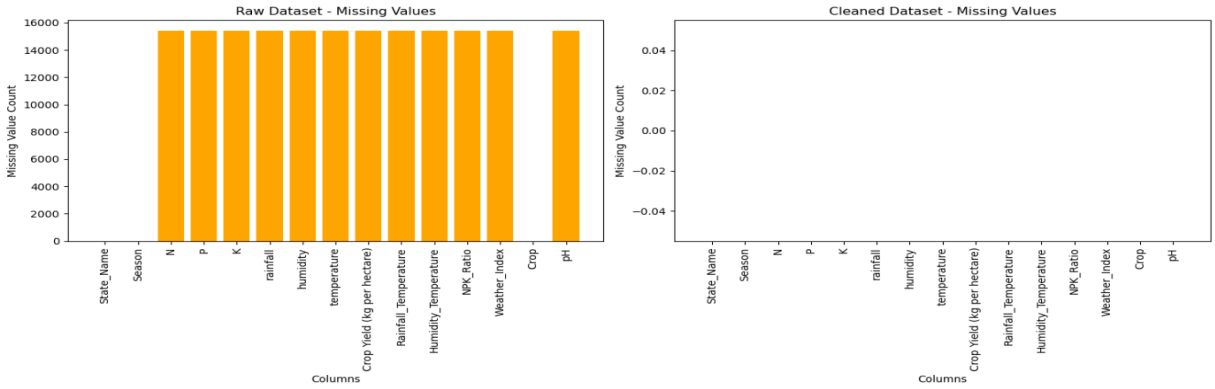

Figure 5.1 Before and After Missing Value Imputation

The above Figure 5.1 shows the comparison of missing values in the raw and cleaned datasets. The raw dataset had significant missing values across multiple features. After preprocessing, all missing values were successfully handled and removed.

5.2.2 Feature Selection Module Screenshot

The Feature Selection Module is responsible for maximizing the machine learning model's performance by determining the most contributory input features and ranking them in their order of influence. The process entails a systematic feature relevance and redundancy analysis to guarantee that only the most significant data attributes are employed during training, hence enhancing model accuracy, interpretability, and efficiency.

One of the most important pieces of this module is the correlation matrix, as it graphically displays the statistical relationships between different numerical features of the dataset. This matrix is calculated based on Pearson correlation coefficients, which estimate the strength of linear association among pairs of continuous variables from -1 (strong negative correlation) to +1 (strong positive correlation), and 0 denotes no linear association.

As shown in Figure 5.2, the module presents a heatmap of the correlation matrix:

The correlation coefficient between each pair of features is represented by each cell in the heatmap (e.g., temperature vs. humidity, rainfall vs. yield, etc.).

Warm colors (e.g., red/orange) are used to represent strong positive correlations and cool colors (e.g., blue) are used to represent strong negative correlations.

Diagonal values are always 1.0 since they are the correlation of a feature with itself.

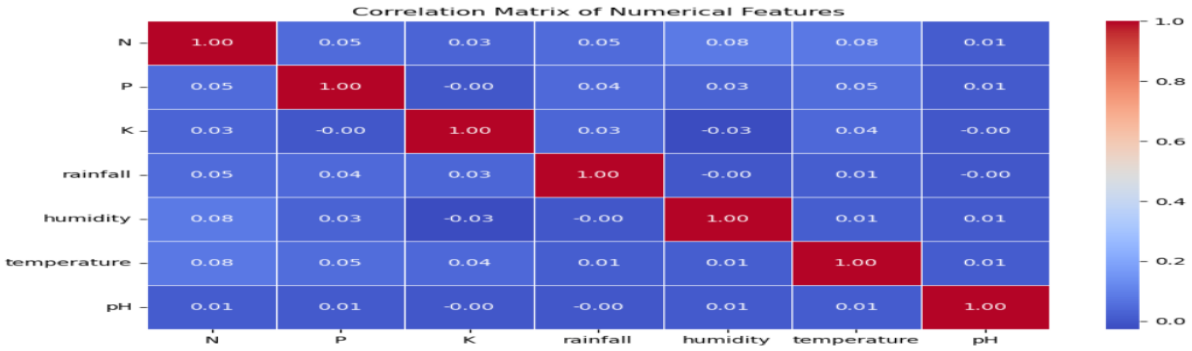

Figure 5.2 Correlation Matrix

The above Figure 5.2 shows the correlation matrix of numerical features used in the crop prediction dataset. Most features have very low correlation with each other, indicating minimal multicollinearity. This suggests that each feature contributes independently to the prediction model.

5.2.3 Model Training Module Screenshot

The Model Training Module is the central part of the system where the machine learning algorithm is trained on the processed dataset to acquire patterns and relationships that facilitate precise predictions. This module processes the cleaned and preprocessed data, chooses the relevant features, and trains the model using supervised learning methods like RF, SVM, or Reinforcement Learning.

During training, the dataset is usually divided into two sets: training data (for training the model) and test data (for testing its performance). The model predicts the target variable from the input features, which in this example consist of

Crop type (classification problem), and

Crop yield (regression problem).

The following is a screenshot in Figure 5.3 demonstrating the results produced after training the model:

For predicting crops, the model returns an accuracy score, which indicates the ratio of correctly predicted crop types to the total number of predictions on the test dataset. The accuracy score is higher when the model is able to predict the correct crop with reliability based on input conditions such as soil nutrients, weather, and season.

To assess performance for predicting yields, the model uses regression measures like R^2 score (coefficient of determination) and Mean Squared Error (MSE). They reflect the measure of how much the predicted yield values are aligned with the real yields. An R^2 score of nearly 1 and low MSE indicate that the model's predictive accuracy is good.

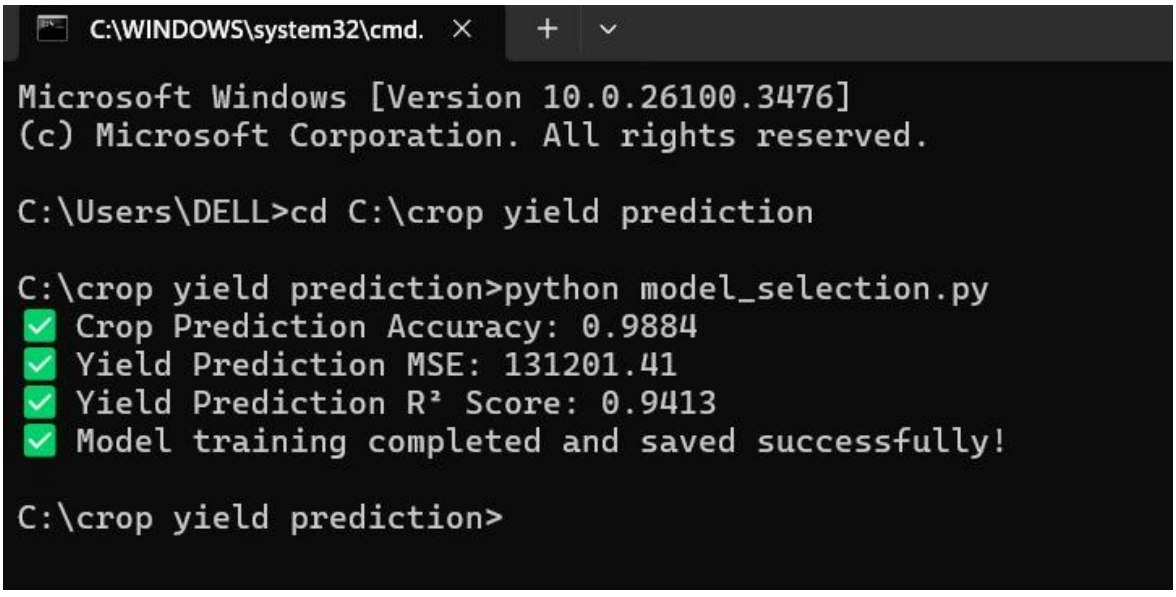

Figure 5.3 Model Accuracy

The above figure 5.3 showcases the evaluation metrics of the crop and yield prediction model. It highlights a high classification accuracy of 98.84% and strong yield prediction performance with an R^2 score of 0.94.

5.2.4 Prediction Module Screenshot

The Prediction Module is the last and most interactive piece of the system, in which users can feed in certain environmental and soil inputs to obtain real-time recommendations for appropriate crop type as well as its probable yield. The module is the deployment of the trained model in an intuitive setting, facilitating real-world decision-making assistance for farmers and agricultural planners.

As shown in Figure 5.4, the interface contains well-labeled input fields for the following parameters

- Nitrogen (N) – amount of nitrogen content in the soil.
- Phosphorus (P) – amount of phosphorus content.
- Potassium (K) – amount of potassium.
- pH – value of soil pH.
- Temperature (°C) – regional average temperature.
- Humidity (%) – atmospheric humidity level.
- Rainfall (mm) – recorded or expected rainfall.
- Season – chosen from a dropdown menu (e.g., monsoon, winter).
- State – the region or location of the user.

After values are inserted, the user submits by clicking the "Predict" button. The system proceeds to infer using the learned machine learning model and shows:

The most suitable crop that matches the inserted conditions.

The calculated crop yield, given in units like kg/ha.

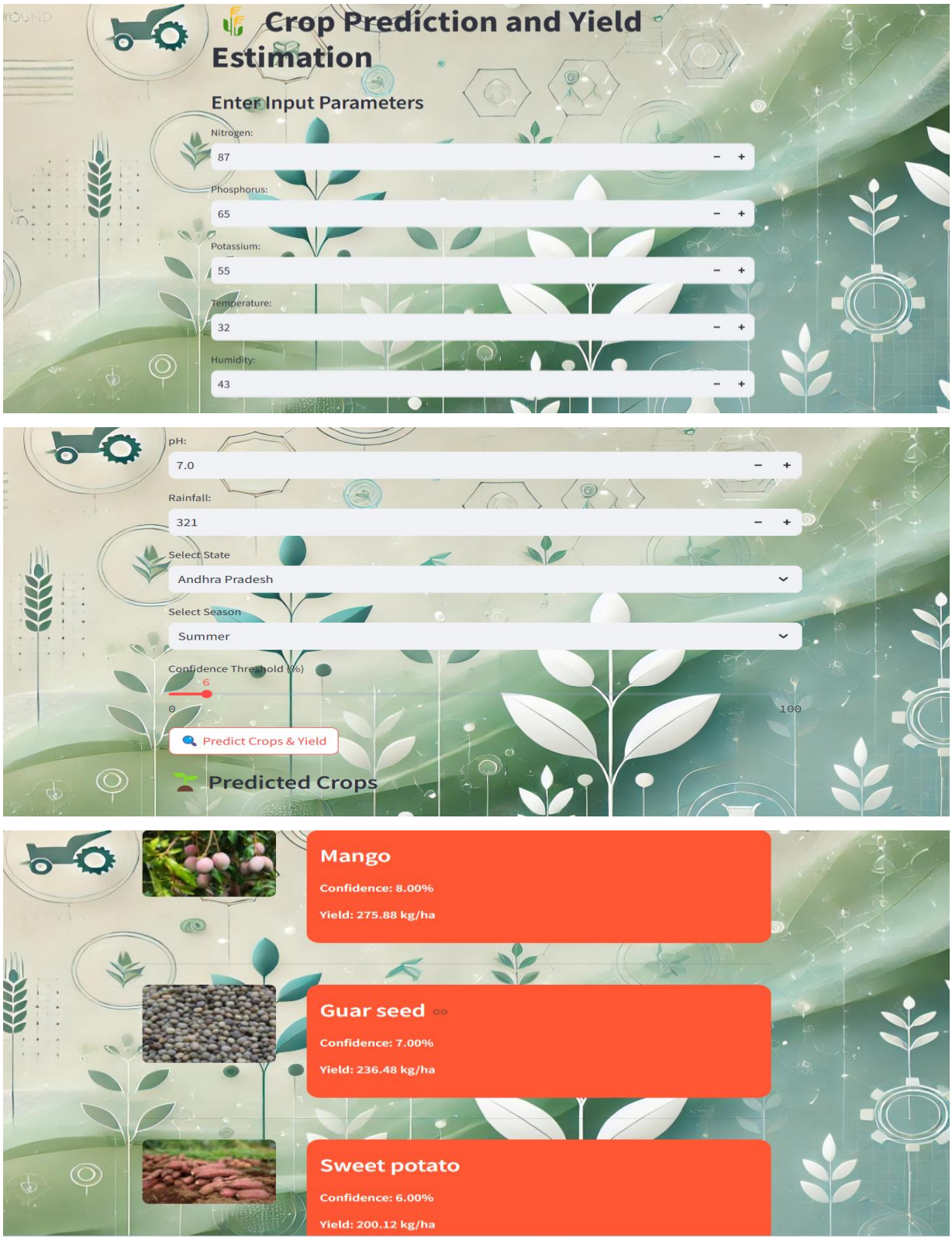

Figure 5.4 Input And Output

The above figure 5.4 illustrates the user input parameters and the resulting crop and yield predictions. It reflects how the model processes environmental and soil data to generate accurate outcomes.

5.3 RESULT ANALYSIS AND DISCUSSION

This section shows the results of the deployed crop prediction system and compares its performance using primary metrics for crop classification and yield estimation.

Accuracy and the confusion matrix are used to evaluate the model's capacity to recognize the most appropriate crop for a particular set of soil and environmental conditions. Accuracy shows the percentage of correctly predicted crop types out of all predictions. Here, the suggested RF model had an accuracy of **98.84%**, reflecting a very high degree of correctness in the identification of suitable crops. The confusion matrix also provides a breakdown of performance over all classes, revealing how frequently each crop was correctly classified or misclassified. It assists in determining which crops the model classifies well and which ones it tends to get confused with, frequently because of shared feature values.

Certain crops, particularly those with more samples in the dataset, for example, rice or wheat, are predicted with greater accuracy. Less popular or region-specific crops, however, have lower accuracy in prediction. This difference often arises from class imbalance, in which some crops are underrepresented in the training data. Another potential reason is feature misrepresentation—if important distinguishing features are missing or not captured well, the model may struggle to distinguish between similar crops. Furthermore, underfitting or overfitting might constrain the model's generalizability on unseen data. Enhancements like SMOTE for the imbalance in the dataset, sophisticated feature engineering, or data augmentation can alleviate these limitations.

For predicting the yield, the performance of the model is measured by regression metrics like the R^2 Score and Mean Squared Error (MSE). These are used to determine how accurate the model's predictions are in relation to actual values of the yield. R^2 measures

the variance in the yield explained by the model, whereas MSE measures the average of the squared difference between actual and predicted values. The suggested RF model had an R^2 value of **0.9413** and an MSE of **131,201.41**, indicating high precision in yield value prediction with relatively low error.

Q-Learning Reinforcement had 75% crop prediction accuracy, an R^2 value of 0.78, and an MSE of 260,000.00.

Support Vector Machine (SVM) had 81% accuracy, an R^2 value of 0.86, and an MSE of 190,000.00.

RF model significantly outperformed both, making it the most effective approach in this study.

The model generally performs better for commonly cultivated crops where more historical data is available. These crops show lower error margins and more consistent predictions. However, for rare crops with limited training data, the model may struggle, leading to larger prediction errors. This is due to the absence of enough and varied samples that reflect the heterogeneity in yield. The addition of more agronomic characteristics such as fertilizer application, pest events, irrigation practices, and seasonal impacts may enhance accuracy. Ensemble techniques or stacking methods may also improve generalizability through integration of the strengths of different algorithms.

In short, the system has exceptional results in both yield prediction and crop classification, with RF model having the best performance. Although it has excellent performance with major crops, improvement in data quality, feature representation, and model tuning may bring more performance to less frequent crops and intricate agricultural situations.

Table 5.1 Prediction Comparison

Algorithm	Crop Prediction Accuracy	Yield Prediction (R^2 Score)	Yield Prediction (MSE)
Q-Learning Reinforcement	75%	0.78	260,000.00
SVM	81%	0.86	190,000.00
RF (Proposed)	98.84%	0.9413	131,201.41

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

This proposed research sought to develop and contrast machine learning systems with the capacity to predict suitable crop types and estimate expected crop yields using primary agricultural indicators. To achieve this, RF Classifiers were used in crop classification tasks, while RF Regressors were used to carry out yield estimation. These models contribute significantly towards advancing precision agriculture by yielding data-driven recommendations.

- Crop Type Prediction: RF-based classification model was found to be reliable, classifying crop types accurately based on parameters such as nitrogen, phosphorus, potassium, temperature, and rainfall level.
- Crop Yield Estimation: RF Regressor functioned effectively in estimating crop yields with little deviation from actual values, demonstrating how climatic conditions and soil nutrients play a pivotal role in production.
- Model Behavior Understanding: Feature importance analysis was conducted for both models to uncover the most influential variables, giving actionable recommendations to allow farmers to prioritize parameters with high impacts.
- Data Preparation and Model Validation: Through data cleaning and transformation processes—such as filling in missing values and scaling inputs—the models were enabled to handle variability in real-world data. Validation methods like accuracy scores, confusion matrices, and residual analysis verified the system's robustness and reliability.

In general, this proposed work finds the promise of machine learning in empowering more intelligent agriculture. From selecting the appropriate crops to predicting yields, such systems can empower farmers with better resource allocation, reduced risks, and improved productivity.

6.2 FUTURE ENHANCEMENTS

Although the present system establishes a solid foundation, several improvements can be undertaken to prolong its efficacy and increase its versatility in various, actual farming settings:

- Data Expansion: Expansion of training data volumes and varieties—through additional inclusion of crop varieties, geographical areas, and diverse climates—can enhance model dependability.
- Enhanced Feature Engineering: Addition of extra input factors such as pest infestations, past yields, real-time weather, and satellite imagery could enhance model performance significantly.
- Alternative Algorithms: Attempting more sophisticated or domain-specific models like XGBoost, SVM, or even deep learning techniques might yield better prediction accuracy.
- Hybrid Models: Applying ensemble learning by integrating various algorithms can assist in better generalizability and overfitting reduction.
- Sensor Integration: Integrating IoT-integrated sensors to gather real-time data (e.g., soil moisture, pH level, ambient temperature) will enable the system to render on-the-go advice.
- Farmer Interface: A user-friendly web or mobile application with live visualizations, dynamic graphs, and prediction updates can greatly enhance usability for end-users.
- Localized Prediction: Building separate models tailored to specific regions will help capture the unique environmental and soil conditions of each region, leading to more precise yield predictions.
- Price Forecasting: By employing time-series modeling techniques such as ARIMA or LSTM, the system can predict future crop prices to help farmers decide on optimal times for selling crops.

- Sustainable Practices: The future models can be designed to include environmental considerations, including water usage, carbon footprint, or soil health.
- Fertilizer Recommendation: Inclusion of a recommendation system to provide recommendations for optimal fertilizer use based on soil and crop conditions can aid in nutrient-balanced management and avoid overuse.
- Smart Chatbot: A virtual assistant powered by AI can be created to assist farmers with advice on farming methods, crop disease, pests, and timely interventions through input data.
- Rotation Strategies: The system can be expanded to suggest crop rotation timetables to maintain soil fertility and control pests efficiently, enhancing short-term and long-term farm production.

REFERENCES

[1] Shreya S. Bhanose, Kalyani A. Bogawar (2020) “Crop And Yield Prediction Model”, International Journal of Advance Scientific Research and Engineering Trends, Volume 1,Issue 1, April 2016

[2] Tripathy, A. K., et al.(2021) “Data mining and wireless sensor network for agriculture pest/disease predictions.” Information and Communication Technologies (WICT), 2011 World Congress on. IEEE.

[3] Ramesh Babu Palepu (2021) ” An Analysis of Agricultural Soils by using Data Mining Techniques”, International Journal of Engineering Science and Computing, Volume 7 Issue No. 10 October.

[4] Rajeswari and K. Arunesh (2020) “Analysing Soil Data using Data Mining Classification Techniques”, Indian Journal of Science and Technology, Volume 9, May.

[5] A.Swarupa Rani (2020), “The Impact of Data Analytics in Crop Management based on Weather Conditions”, International Journal of Engineering Technology Science and Research, Volume 4,Issue 5,May.

[6] Pritam Bose, Nikola K. Kasabov (2020), “Spiking Neural Networks for Crop Yield Estimation Based on Spatiotemporal Analysis of Image Time Series”, IEEE Transactions On Geoscience And Remote Sensing.

[7] Priyanka P.Chandak (2021),” Smart Farming System Using Data Mining”, International Journal of Applied Engineering Research, Volume 12, Number 11.

[8] Vikas Kumar, Vishal Dave (2021), “KrishiMantra: Agricultural Recommendation System”, Proceedings of the 3rd ACM Symposium on Computing for Development, January.

[9] Savae Latu (2021), ”Sustainable Development : The Role Of Gis And Visualisation”, The Electronic Journal on Information Systems in Developing Countries, EJISDC 38, 5, 1-17.

[10] Nasrin Fathima.G (2020), “Agriculture Crop Pattern Using Data Mining Techniques”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, May.

[11] Ramesh A.Medar (2020), ”A Survey on Data Mining Techniques for Crop Yield Prediction”, International Journal of Advance Research in Computer Science and Management Studies, Volume 2, Issue 9, September.

[12] Shakil Ahamed.A.T.M, Navid Tanzeem Mahmood (2021),” Applying data mining techniques to predict annual yield of major crops and recommend planting different crops in different districts in Bangladesh”, ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD),IEEE,June.

[13] Shreya S.Bhanose (2020),”Crop and Yield Prediction Model”, International Journal of Advence Scientific Research and Engineering Trends, Volume 1,Isssue 1,ISSN(online) 2456- 0774,April.

[14] Agaj I Iorshase, Onyeke Idoko Charles,”A Well-Built Hybrid Recommender System for Agricultural Products in Benue State of Nigeria”, Journal of Software Engineering and Applications,2021,8,581-589.

[15] G. Adomavicius and A. Tuzhilin(2020), “Toward the Next Generation of Recommender Systems: A Survey of the State-of-theArt and Possible Extensions,” IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June.

APPENDIX

A1-SOURCE CODE

MODEL CODE

```
import pandas as pd

import numpy as np

import pickle

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, mean_squared_error, r2_score

# ☑ Step 1: Load and Clean the Dataset

df = pd.read_csv("crop_yield_balanced_with_ph.csv")

# ☑ Step 2: Remove Bias-Causing Features

drop_features = ["Rainfall_Temperature", "Weather_Index", "Humidity_Temperature"]

df.drop(columns=drop_features, inplace=True, errors="ignore")

# ☑ Step 3: Define Features and Target Variables

numerical_features = ["N", "P", "K", "rainfall", "humidity", "temperature", "pH"]

categorical_features = ["State_Name", "Season"]

target_crop = "Crop"

target_yield = "Crop Yield (kg per hectare)"
```

```
# ☑ Encode Crop Type and Add It to Yield Features
df[“Crop_Code”] = df[“Crop”].astype(“category”).cat.codes  # Convert Crop to Numeric
# ☑ Step 4: Preprocessing (Scaling + One-Hot Encoding) **without Crop_Code**
preprocessor = ColumnTransformer([
    (“num”, StandardScaler(), numerical_features),  # Exclude Crop_Code
    (“cat”, OneHotEncoder(handle_unknown=”ignore”), categorical_features)])
# Apply transformations
X_crop = preprocessor.fit_transform(df.drop(columns=[target_crop, target_yield, “Crop_Code”]))  # Exclude Crop_Code for crop model
y_crop = df[target_crop]
# ☑ Save Preprocessor for Use in Streamlit App
with open(“preprocessor.pkl”, “wb”) as f:
    pickle.dump(preprocessor, f)
# ☑ Step 5: Train-Test Split
X_train, X_test, y_crop_train, y_crop_test = train_test_split(X_crop, y_crop, test_size=0.2, random_state=42)
# ☑ Step 6: Train Crop Prediction Model
crop_model = RandomForestClassifier(n_estimators=100, random_state=42)
crop_model.fit(X_train, y_crop_train)
# ☑ Step 7: Train Yield Prediction Model (Now Uses Crop Type)
df[“Crop_Code”] = df[“Crop”].astype(“category”).cat.codes  # Ensure Crop_Code is set
```

```
numerical_features.append(“Crop_Code”)  # Add Crop_Code only for yield model

preprocessor_yield = ColumnTransformer([

    (“num”, StandardScaler(), numerical_features),

    (“cat”, OneHotEncoder(handle_unknown=”ignore”), categorical_features)])

X_yield = preprocessor_yield.fit_transform(df.drop(columns=[target_crop, target_yield]))
# Now include Crop_Code

y_yield = df[target_yield]

X_train_yield, X_test_yield, y_yield_train, y_yield_test = train_test_split(X_yield, y_yield,
test_size=0.2, random_state=42)

yield_model = RandomForestRegressor(n_estimators=100, random_state=42)

yield_model.fit(X_train_yield, y_yield_train)

# ☑ Save the Second Preprocessor (for Yield)

with open(“preprocessor_yield.pkl”, “wb”) as f:

    pickle.dump(preprocessor_yield, f)

# ☑ Save Models

with open(“crop_model.pkl”, “wb”) as f:

    pickle.dump(crop_model, f)

with open(“yield_model.pkl”, “wb”) as f:

    pickle.dump(yield_model, f)

print(“☑ Models trained and saved successfully!”)
```

APP CODE

```
import streamlit as st
import pickle
import numpy as np
import pandas as pd
import base64
import os
# Function to find crop images
def find_image(crop_name, image_folder="crop_images"):
    extensions = [".jpg", ".jpeg", ".png"]
    for ext in extensions:
        image_path = os.path.join(image_folder, crop_name + ext)
        if os.path.exists(image_path):
            return image_path
    return None
# Cache model loading
@st.cache_resource
def load_models():
    with open("crop_model.pkl", "rb") as f:
        crop_model = pickle.load(f)
    with open("yield_model.pkl", "rb") as f:
```

```
        yield_model = pickle.load(f)
    with open("preprocessor.pkl", "rb") as f:
        preprocessor = pickle.load(f)
    with open("preprocessor_yield.pkl", "rb") as f:
        preprocessor_yield = pickle.load(f)
    return crop_model, yield_model, preprocessor, preprocessor_yield
crop_model, yield_model, preprocessor, preprocessor_yield = load_models()


crop_code_mapping = {
    'Arecanut': 0, 'Arhar_tur': 1, 'Ash Gourd': 2, 'Bajra': 3, 'Banana': 4, 'Barley': 5, 'Bean': 6,
    'Beans and Mutter(Vegetable)': 7, 'Beet Root': 8, 'Ber': 9, 'Bhindi': 10, 'Bitter Gourd': 11,
    'Black pepper': 12, 'Blackgram': 13, 'Bottle Gourd': 14, 'Brinjal': 15, 'Cabbage': 16,
    'Cardamom': 17, 'Carrot': 18, 'Cashewnut': 19, 'Castor seed': 20, 'Cauliflower': 21,
    'Citrus Fruit': 22, 'Coconut': 23, 'Colocosia': 24, 'Coriander': 25, 'Cotton(lint)': 26,
    'Cowpea(Lobia)': 27, 'Cucumber': 28, 'Drum Stick': 29, 'Dry chillies': 30, 'Dry ginger': 31,
    'Garlic': 32, 'Ginger': 33, 'Gram': 34, 'Grapes': 35, 'Groundnut': 36, 'Guar seed': 37,
    'Horse-gram': 38, 'Jack Fruit': 39, 'Jobster': 40, 'Jowar': 41, 'Jute': 42, 'Kapas': 43,
    'Khesari': 44, 'Korra': 45, 'Lab-Lab': 46, 'Lemon': 47, 'Lentil': 48, 'Linseed': 49,
```

‘Maize’: 50, ‘Mango’: 51, ‘Masoor’: 52, ‘Mesta’: 53, ‘Moong(Green Gram)’: 54, ‘Moth’: 55,

‘Niger seed’: 56, ‘Onion’: 57, ‘Orange’: 58, ‘Paddy’: 59, ‘Papaya’: 60, ‘Peas and beans (Pulses)’: 61,

‘Pineapple’: 62, ‘Pome Fruit’: 63, ‘Pome Granet’: 64, ‘Potato’: 65, ‘Pump Kin’: 66,

‘Ragi’: 67, ‘Rajmash Kholar’: 68, ‘Rapeseed and Mustard’: 69, ‘Redish’: 70,

‘Ribed Guard’: 71, ‘Rice’: 72, ‘Ricebean (nagadal)’: 73, ‘Rubber’: 74, ‘Safflower’: 75,

‘Samai’: 76, ‘Sannhamp’: 77, ‘Sapota’: 78, ‘Sesamum’: 79, ‘Small millets’: 80,

‘Snak Guard’: 81, ‘Soyabean’: 82, ‘Sugarcane’: 83, ‘Sunflower’: 84, ‘Sweet potato’: 85,

‘Tapioca’: 86, ‘Tea’: 87, ‘Tobacco’: 88, ‘Tomato’: 89, ‘Turmeric’: 90,

‘Urad’: 91, ‘Varagu’: 92, ‘Water Melon’: 93, ‘Wheat’: 94, ‘Perilla’: 95, ‘Yam’: 96

}

state_options = [“Andhra Pradesh”, “Arunachal Pradesh”, “Assam”, “Bihar”, “Chhattisgarh”,

“Goa”, “Gujarat”, “Haryana”, “Himachal Pradesh”, “Jharkhand”, “Karnataka”,

“Kerala”, “Madhya Pradesh”, “Maharashtra”, “Manipur”, “Meghalaya”,

“Mizoram”, “Nagaland”, “Odisha”, “Punjab”, “Rajasthan”, “Sikkim”,

“Tamil Nadu”, “Telangana”, “Tripura”, “Uttar Pradesh”, “Uttarakhand”,

“West Bengal”]

season_options = [“Kharif”, “Rabi”, “Summer”, “Whole Year”, “Winter”]

Set background image

```
def set_background():
    image_path = “background.jpg”
    if os.path.exists(image_path):
        with open(image_path, “rb”) as img:
            encoded_img = base64.b64encode(img.read()).decode()
        bg_style = f”””
            <style>
                [data-testid=”stAppViewContainer”] {{
                    background: url(“{encoded_img}”) !important;
                    background-size: cover !important;
                    background-position: center !important;
                    background-attachment: fixed !important;
                }}
            </style>
        “””
        st.markdown(bg_style, unsafe_allow_html=True)

set_background()

# Page title
st.title(“ 🌾 Crop Prediction and Yield Estimation”)
```

```
# Input fields (moved from sidebar to center)
st.subheader("Enter Input Parameters")
filtered_crops = {}
VALID_RANGES = {
    "Nitrogen": (10, 120),
    "Phosphorus": (10, 120),
    "Potassium": (10, 120),
    "Temperature": (10, 60),
    "Humidity": (10, 100),
    "pH": (3.5, 9.0),
    "Rainfall": (0, 1000)
}

# Collect User Inputs
inputs = {}
for label, (min_val, max_val) in VALID_RANGES.items():
    step = 0.1 if isinstance(min_val, float) else 1
    inputs[label] = st.number_input(f"{label}:", step=step, format="%.1f" if step == 0.1 else "%d")
state = st.selectbox("Select State", state_options)
```

```
season = st.selectbox(“Select Season”, season_options)

confidence_threshold = st.slider(“Confidence Threshold (%)”, min_value=0, max_value=100, value=10)

# Predict Button

if st.button(“🔍 Predict Crops and Yield”):

    error_messages = []

    # **Validate Inputs**

    for label, (min_val, max_val) in VALID_RANGES.items():

        if not (min_val <= inputs[label] <= max_val):

            error_messages.append(f”❌ <strong>{label}</strong> must be between <strong>{min_val} – {max_val}</strong>.”)

    # **If Errors Exist, Show Messages and Stop**

    if error_messages:

        error_message = “<br>”.join(error_messages)

        st.markdown(

            f”””

            <div style=”

                background-color: #FFDAB9;  /* Light Orange */

                color: #D35400;  /* Dark Orange */

                font-size: 18px;

                font-weight: bold;

                padding: 15px;
```

```
            border-radius: 10px;
            border: 3px solid #D35400;
            box-shadow: 3px 3px 10px rgba(0,0,0,0.3);
        ">
            ⛔ **Invalid Inputs!** Please correct the following:
            <br>{error_message}
        </div>
        """,
        unsafe_allow_html=True
    )
    st.stop()  # **⛔ Stop Execution Immediately**

# **If Inputs are Valid, Proceed with Prediction**
df_test = pd.DataFrame([[inputs["Nitrogen"], inputs["Phosphorus"], inputs["Potassium"],
                         inputs["Temperature"], inputs["Humidity"], inputs["pH"],
                         inputs["Rainfall"], state, season]],
                       columns=['N', 'P', 'K', 'temperature', 'humidity', 'pH', 'rainfall', 'State_Name', 'Season'])

# **Preprocess and Predict Crops**
X_crop = preprocessor.transform(df_test)
crop_probs = crop_model.predict_proba(X_crop)[0]
```

```
crop_predictions = {crop: prob * 100 for crop, prob in zip(crop_code_mapping.keys(), crop_probs)}

crop_predictions = {k: v for k, v in sorted(crop_predictions.items(), key=lambda item: item[1], reverse=True)}

# **Filter Crops Based on Confidence Threshold (user-defined)**

filtered_crops = {crop: confidence for crop, confidence in crop_predictions.items() if confidence >= confidence_threshold}

# **If No Crops Exceed Confidence Threshold, Stop Execution**

if not filtered_crops:

    st.warning(f"⃠ No crops exceed the {confidence_threshold}% confidence threshold.")

    st.stop()

# **Yield Prediction and Results Table**

df_yield = pd.DataFrame(columns=["#", "Image", "Crop", "Confidence (%)", "Predicted Yield (kg/ha)"])

for idx, (crop, confidence) in enumerate(filtered_crops.items(), start=1):

    image_path = find_image(crop)  # Find image for each crop


    df_test["Crop_Code"] = crop_code_mapping.get(crop, -1)

    X_yield = preprocessor_yield.transform(df_test)

    base_yield = yield_model.predict(X_yield)[0]

    predicted_yield = base_yield * (confidence / 100)
```

```
        df_yield.loc[len(df_yield)] = [idx, image_path if image_path else “No Image”, crop, f”{confidence:.2f}%”, f”{predicted_yield:.2f} kg/ha”]

    # **Display Results**

    st.subheader(“🌱 **Predicted Crops**”)


    for I in range(len(df_yield)):

        confidence = float(df_yield.iloc[i][‘Confidence (%)’].replace(‘%’, ‘’))


        # **Determine Background Color Based on Confidence Level**

        bg_color = “#4CAF50” if confidence >= 70 else “#FFC107” if confidence >= 40 else “#FF5733”


        with st.container():

            st.markdown(“---")  # **Separator**


            col1, col2 = st.columns([1, 3])


            # **Image Section**

            with col1:

                image_path = df_yield.iloc[i][‘Image’]

                if image_path and os.path.exists(image_path):

                    st.image(image_path, width=150)
```

```
        else:
            st.warning("⃠ No Image Available")


    # **Crop Details Section**
    with col2:
        st.markdown(
            f"""
            <div style="background-color: {bg_color}; padding: 15px; border-radius: 15px; color: white;">
                <h3 style="margin-bottom: 5px;">{df_yield.iloc[i]['Crop']}</h3>
                <p><strong>Confidence: {df_yield.iloc[i]['Confidence (%)']}</strong></p>
                <p><strong>Yield: {df_yield.iloc[i]['Predicted Yield (kg/ha)']}</strong></p>
            </div>
            """,
            unsafe_allow_html=True
        )
```

A2-SCREENSHOTS

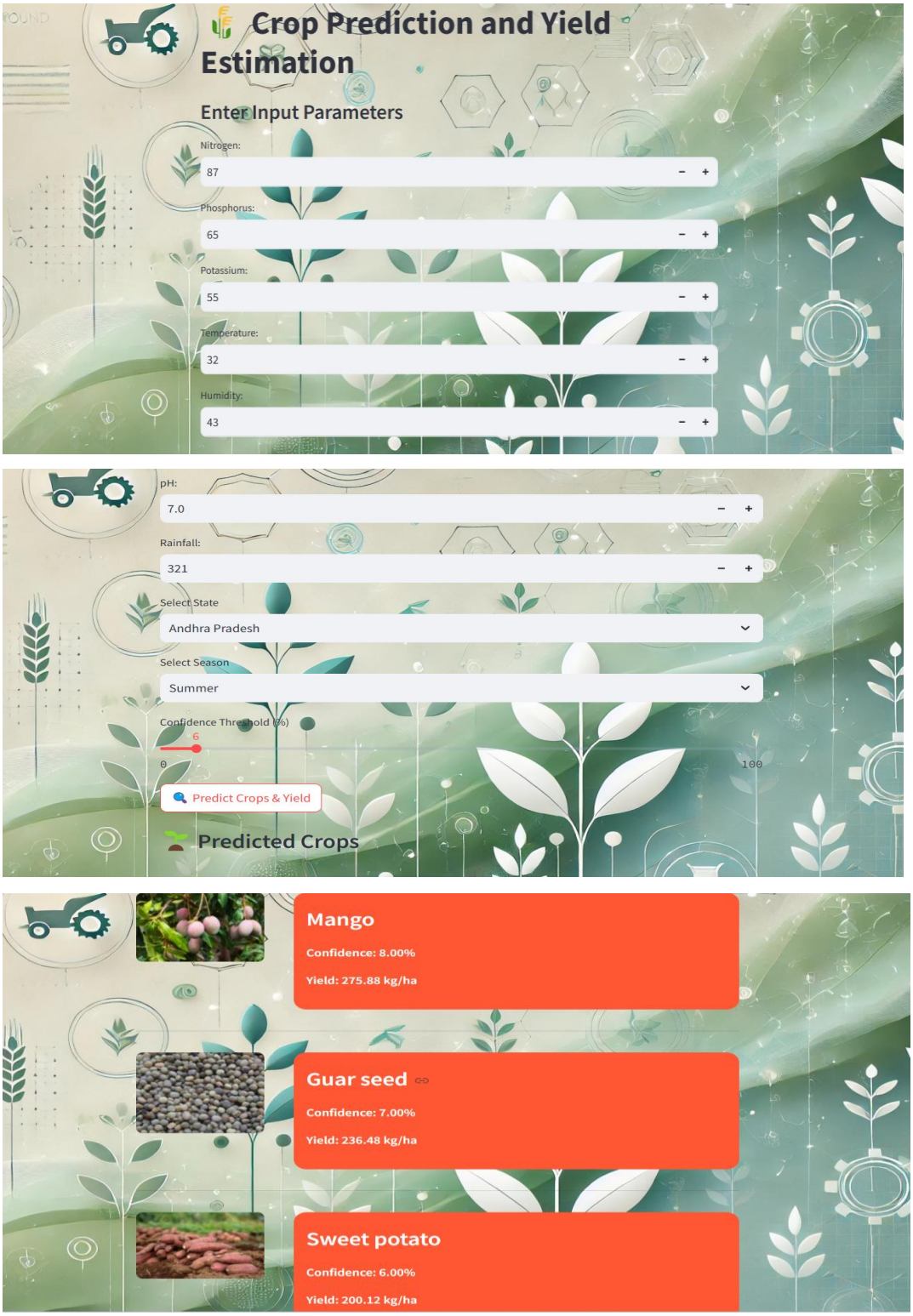

Figure A2.1 Output Screenshots

The above Figure A2.1 presents the final output interface of the crop prediction and yield estimation system, displaying the predicted crop names along with their estimated yield in kilograms per hectare. The screenshots highlight the effectiveness of the model in interpreting user inputs such as soil nutrients, temperature, humidity, and other environmental factors to generate accurate and practical recommendations. The system's high prediction accuracy and intuitive interface make it suitable for real-world agricultural decision-making, enhancing productivity and guiding farmers with data-driven insights.

TECHNICAL BIOGRAPHY

MOHAMED HASIN F. (210071601125) was born in Virudhachalam, Tamilnadu, India in 2003. He completed high school HSC certification in 2021. He is currently pursuing a B.Tech degree in Computer Science and Engineering (CSE) at the School of Computer Information and Mathematical Sciences at B.S. Abdur Rahman Crescent Institute of Science and Technology, which is located at Vandalur, Chennai, India. He has a strong interest in the field of Networks, Data Science, and Machine learning. The Email for communication is hasinace10@gmail.com and his contact number is +91 9597128066

MOHAMED IMAD MASOOD T A (210071601097) was born in Kadayanallur, Tamil Nadu, India in 2003.He completed high school HSC certification in 2021. He is currently pursuing a B.Tech degree in Computer Science and Engineering (CSE) at the School of Computer Information and Mathematical Sciences at B.S. Abdur Rahman Crescent Institute of Science and Technology, which is located at Vandalur, Chennai, India. He has a strong interest in the field of Mathematics, Data Science, Networks and Machine learning. The Email for communication is imadmasood1234@gmail.com and his contact number is +91 8248907802

Imad Masood

RE-2022-540501

Batch 2

Batch 2

Universidad del Valle

Document Details

Submission ID **trn:oid:::26066:449953428**

Submission Date

Apr 17, 2025, 12:36 PM GMT+5:30

Download Date

Apr 17, 2025, 12:41 PM GMT+5:30

File Name

RE-2022-540501.pdf

File Size

832.0 KB

44 Pages

9,976 Words

60,140 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

112 Not Cited or Quoted 10%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
5% Publications
7% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.