ELSEVIER

# JOURNAL ON COMMUNICATIONS

ISSN:1000-436X

Scopus

REGISTERED

www.jocs.review

# ML CROP SELECTION AND YEILD FORCASTING

Prof. Madhavi Sadu[1] Mr. Shubham Nanne[2] Mr. Swayam Bommewar[3] Ms. Pooja Prajapati[4] Ms. Kashish Gorghate[5] Mr. Mohan Lavhale[6]

Assistant professor
RCERT Chandrapur
Student
RCERT Chandrapur

***Abstract:*** *: Agriculture is very important to the world's economy; picking the right crops—and knowing how much they will produce—helps ensure there is enough food and that farms run efficiently. Traditionally, farmers make these decisions based on experience, which can take a lot of time and isn't always accurate. The paper focuses on the design of an intelligent ML-based system capable of recommending the best crops to grow and predicting their yield. In recommending suitable crops for plantation, the soil type, pH value of the soil, atmospheric temperature, rainfall, and humidity are among the data considered. Guided by machine learning algorithms such as Random Forest, Decision Trees, and Gradient Boosting, the system evaluates all these elements in past data, including current environmental conditions, to propose the best crop for the farm. The prediction of yield production represents the quantity of crops the system expects to be produced. It analyses past yield records, weather patterns, and other farming factors. Methods such as Linear Regression, XGBoost, and Support Vector Regression enable the system to understand how different conditions influence crop output. In this way, farmers can do better planning of irrigation and resources and plan marketing strategies. This system is user-friendly since a farmer only needs to input all his data into the system and, within a very short time, suggestions and predictions appear. Its accuracy was checked using performance measures such as accuracy, mean squared error (MSE), and R-squared. The early results indicate that the recommendations and predictions made using this approach were better compared to the traditional approaches.*

***Keywords: Machine Learning, Crop Recommendation, Yield Prediction, Precision Agriculture, Random Forest, Decision Trees, Gradient Boosting, Linear Regression, XGBoost, Support Vector Regression, Agricultural Data Analysis, Climate-based Crop Selection Farm Productivity Soil and Weather Parameters***.

## 1. INTRODUCTION

### 1.1 The Need for Intelligent Agricultural Systems

Agriculture is indispensable in feeding the world's population and maintaining economic stability. Selection of appropriate crops, together with yield estimation, is very crucial in farmers' decisions on how to maximize production. Traditionally, this has been done based on a farmer's experience and history, which tends to be subjective, slow, and sometimes inaccurate. Climate change, soil degradation, and unpredictable weather patterns make farming decisions more difficult in modern times.

But with an increasing availability of agricultural and environmental data, along with important strides in machine learning, comes a chance to develop intelligent systems offering farmers dependable, data-driven recommendations. This could improve efficiency and reduce uncertainty for better decision-making in agriculture.

## 1.2 Evolution of Crop Recommendation and Yield Prediction Systems

### 1.2.1 Rule-Based Approaches

The early automated farming advisory tools were simple, rule-based systems. These systems applied simple decision trees or expert-system logic to recommend crops or estimate yields. While such systems reduced the amount of guesswork required, they could not consider complex environmental factors or conditions that changed rapidly, thus limiting their accuracy.

### 1.2.2 Data-Driven and Statistical Methods

The next series of statistical models used by researchers involved linear regression, multiple regression, and time-series analysis. These approaches made it possible to predict yields based on historical soil and climate data. Although offering measurable insights, they struggled with non-linear and complex relationships common in agriculture and required large and quality datasets to perform well.

### 1.2.3 Machine Learning and Ensemble Techniques

More recently, machine learning models such as Random Forest, Gradient Boosting, and XGBoost have significantly improved crop and yield predictions. These models analyze huge datasets, capture nonlinear patterns, and, in general, give more accurate and personalized recommendations. Consequently, ML-based systems can present farmers with more accurate crop suggestions and better yield forecasts than previous approaches.

## 1.3 Theoretical and Technological Foundations

### 1.3.1 Machine Learning for Agriculture

Machine learning identifies patterns and relationships in complex datasets related to agriculture. Crop recommendation systems use methodologies of supervised learning to find the best crops that can be grown on soils of a particular type, pH, rainfall, temperature, and other factors. Yield prediction models use regression algorithms to estimate the amount of crop output that can be expected by analyzing past yield data along with weather conditions and other agronomic features.

### 1.3.2 Development Tools and Frameworks

With modern ML tools like Scikit-learn and XGBoost, it is easier now to build and evaluate a prediction model. Other libraries, such as Pandas and NumPy, assist with data cleaning, analysis, and visualization. On top of that, web frameworks such as Flask or Streamlit enable the models to be converted into user-friendly applications, so that farmers can input their data and receive recommendations instantly.

### 1.3.3 Integrated System Architecture

A comprehensive intelligent agriculture would involve data collection, preprocessing, model training, predictions, and visualizations of outputs in a user-friendly format. Reliable crop recommendations and yield forecasts could only be achieved with accurate environmental data and robust machine learning models, especially ensemble methods.

## 2. OBJECTIVES

The main goal of this project is to leverage machine learning techniques to assist farmers and agricultural planners in making informed decisions that can enhance crop productivity and optimize resource utilization. The specific objectives are as follows:

### 2.1 Develop a Crop Recommendation System:

o Analyzing environmental and soil parameters such as soil type, pH, temperature, rainfall, and humidity.

o Recommend the most suitable crops to grow on a specific piece of land based on historic and real-time agricultural data.

o Reduce dependence on traditional, experience-based methods of crop selection.

### 2.2 Implement Yield Forecasting Models:

- Predict expected crop yields using historical yield data, climatic conditions, and agronomic factors.
- Utilize regression-based and ensemble machine learning algorithms such as Linear Regression, XGBoost, and Support Vector Regression.
- Enable farmers to plan resources, irrigation.

### 2.3 Create a User-Friendly Interface:

- Design a user-friendly web interface where farmers can input the relevant parameters.
- Provide instant crop recommendations and yield predictions in simple-to-understand formats.

### 2.4 Ensure Model Accuracy and Reliability:

- Validate models using performance metrics like accuracy, mean squared error (MSE), and R-squared value.
- Continuously improve predictive capability to provide robust and dependable results.

### 2.5 Promote Sustainable and Efficient Agriculture:

- Support optimized use of land, water, and other resources.
- Reduce crop failure risks and increase overall farm profitability.
- Contribute to data-driven, sustainable farming practices for long-term agricultural growth.

## 3. LITERATURE REVIEW

### 3.1 Early Approaches in Crop Recommendation and Yield Prediction

Previously, farmers usually chose crops based on their experience and estimated yields; although this worked to some degree, the process was fairly slow, subjective, and not very reliable, especially under unpredictable weather conditions and changing soil conditions.

Early researchers tried to enhance this by developing rule-based systems that suggested the appropriate crops based on basic factors such as soil type, rainfall, and temperature. This helped a little, but these couldn't handle complex situations where many factors interacted with each other.

Later, other statistical methods of predicting crop yield from historical data included linear regression and time-series forecasting. These were more advanced but still limited, as they assumed the relationship between factors was always linear, although real agricultural conditions usually are far more complex.

### 3.2 Machine Learning Applications in Agriculture

Machine learning has enhanced crop recommendations and yield predictions significantly. Unlike earlier methods, ML can analyze huge amounts of data, discover hidden patterns, and understand nonlinear relationships.

These algorithms are widely used, such as Random Forest, Decision Trees, Gradient Boosting, and XGBoost, since they give more accurate and reliable results.

### 3.3 Integrated Crop Recommendation and Yield Forecasting Systems

Some researchers have also designed the systems that perform both tasks: recommend the best crop and estimate its yield.

For example, a system proposed by Singh et al. 2020 has two modules, one of which recommends crops based on environmental data and the other predicts yield using regression models. This will help farmers plan better and reduce the risk of crop failure.

The use of ensemble techniques-whereby multiple models are combined-also enhances the accuracy. A research study by Sharma & Joshi, 2021 has combined Random Forest and XGBoost and obtained better results than using either of these two models alone.

### 3.4 Role of Environmental and Climatic Data

Soil and climate data are crucial for accurate predictions. Important factors include:

- Soil texture and fertility
- pH levels
- Rainfall
- Temperature
- Humidity

Systems that combine real-time climate data with historical yield records perform much better. Satellite images and remote sensing are also being used to monitor fields more accurately and improve predictions.

### 3.5 Challenges and Limitations in Existing Research

Even with all these improvements, several problems still exist:

Poor or incomplete data

Models not working well in new regions because conditions vary widely

Lack of user-friendly tools that farmers can easily understand and use

There's a need for systems that are accurate **and** practical for everyday farming.

**3.6 Recent Advances and Technological Foundations**
Recent research combines ML techniques with interactive platforms. Web and mobile applications provide farmers with the ability to input data from their farms and receive recommendations in real time. Kumar et al. (2023) illustrate that the integration of ML models with frameworks such as Flask or Streamlit enhances usability, with models that can update continuously with the inclusion of newer data. Hybrid approaches, which integrate regression, ensemble learning, and even deep learning, are being developed to further improve both crop recommendation and yield prediction accuracy.

## 4. PROPOSED SYSTEM

### 4.1 System Overview

The proposed **ML-based Crop Recommendation and Yield Forecasting System** integrates advanced machine learning algorithms with a user-friendly interface to provide actionable guidance to farmers. This system addresses the limitations of traditional agricultural planning by combining **crop recommendation** and **yield prediction** in a single, cohesive platform. The architecture builds upon established agricultural ML research while introducing a modular, dual-function design that improves accuracy, usability, and real-time responsiveness.

### 4.2 System Architecture

#### 4.2.1 Overall Architecture Design

The system follows a **multi-layered architecture** consisting of three main components:

1. **Frontend Interface Layer:** Provides a responsive web or mobile interface for farmers to input environmental and farm data.
2. **Backend Processing Layer:** Handles business logic, ML model integration, and data management.
3. **Data Intelligence Layer:** Manages the ML models, historical yield data, soil and weather databases, and prediction outputs.

This layered approach ensures modularity, scalability, and maintainability, allowing independent development and testing of each module.

#### 4.2.2 Architectural Flow

The system follows a structured workflow:

- Users input farm data such as soil type, pH, temperature, rainfall, and previous crop information via the frontend interface.

- The backend validates and preprocesses the input data.
- The **Crop Recommendation Engine** analyzes environmental parameters using Random Forest, Decision Tree, and Gradient Boosting models to suggest suitable crops.
- The **Yield Forecasting Engine** predicts expected production using regression-based models including Linear Regression, XGBoost, and Support Vector Regression.
- Results are aggregated and presented to the user through an intuitive dashboard, including suggested crops, predicted yield, and resource optimization tips.

### 4.3 Component Architecture

#### 4.3.1 Frontend Architecture

The user interface implements a simple, interactive design. Key features include:

- Data input forms for soil, climate, and farm information
- Dashboard for visualizing crop recommendations and yield predictions
- Real-time updates for immediate feedback

#### 4.3.2 Backend Architecture

The backend uses **Python and Flask** to implement RESTful APIs:

- Data validation and preprocessing
- Integration with ML models for crop recommendation and yield forecasting
- Management of historical datasets and environmental data

#### 4.3.3 Crop Recommendation Engine

This engine evaluates farm and environmental parameters using **Random Forest, Decision Trees, and Gradient Boosting** to identify crops best suited for local conditions. The system leverages historical data and climatic trends to improve recommendation accuracy.

#### 4.3.4 Yield Forecasting Engine

The yield prediction module uses **Linear Regression, XGBoost, and Support Vector Regression** to estimate crop productivity. Input features include soil parameters, weather data, previous crop yields, and agronomic practices.

#### 4.3.5 Data Management and Knowledge Base

A structured database stores:

- Historical crop yields
- Soil and climate data
- Crop characteristics and resource requirements

Automated validation, version control, and backup protocols ensure data integrity and consistency.

### 4.4 System Integration

#### 4.4.1 Integration Architecture

The system integrates all components through intelligent workflow management:

- User input → Data preprocessing → Parallel execution of crop recommendation and yield prediction → Aggregation of results → Dashboard display.

#### 4.4.2 Deployment Architecture

The system can be deployed as a **web or mobile application**, with scalable cloud hosting. Considerations include:

- Stateless design for horizontal scaling
- Efficient caching for frequently used queries
- Database connection pooling for resource optimization

### 4.5 System Innovation and Advantages

This dual-module system represents a significant advancement in agricultural decision-support:

- Provides **accurate, data-driven crop recommendations**
- Predicts **expected yield** to optimize farm planning and resource management Integrates historical, environmental, and real-time data in one

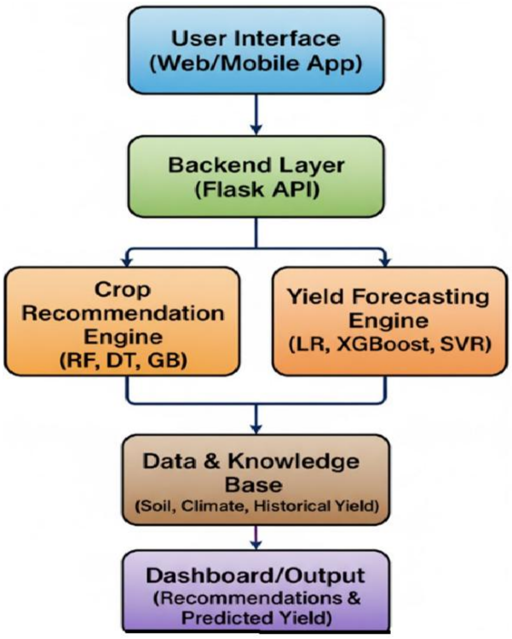



*Figure 1 : Dataflow Diagram*

### 4.6 Accuracy comparison

Table.1 offers the accuracy of crop advice responsibilities across diverse algorithms . This desk unequivocally demonstrates that the random wooded area set of rules surpasses all others, accomplishing a extraordinary accuracy of 99.54%.

| Algorithm | Accuracy |
|---|---|
| Decision Tree | 90.0 |
| Gaussian Naive Bayes | 99.09 |
| Support Vector Machine (SVM) | 10.68 |
| Logistic Regression | 95.23 |
| Random Forest | 99.55 |
| XGBoost | 99.09 |
| KNN | 97.50 |

*Table 1: Algorithm vs Accuracy*

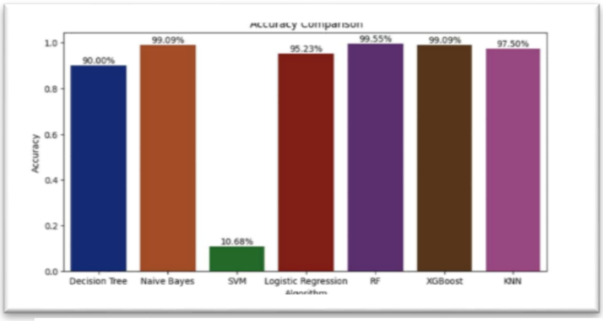



*Figure 2: Algorithm vs Accuracy Bar chat*

## 5.METHODOLOGY

### 5.1 Development Approach and Project Lifecycle

For this project, we adopted an **Agile development methodology**, which allowed us to work in iterative sprints. This approach enabled continuous feedback and frequent improvements, ensuring the project aligned with evolving requirements. The development process was divided into five key phases:

1. **Requirement Analysis and Planning:** We began by identifying the critical factors affecting crop yield, including soil type, weather patterns, crop variety, fertilizer usage, and irrigation practices. We also defined the system specifications based on these factors.

2. **Prototype Development and Core Data Flow Implementation:** Next, we focused on building a prototype with the core data flow. This phase involved setting up the basic pipeline for crop yield predictions and developing a user-friendly interface for data input.

3. **Machine Learning Model Integration and Training:** In this phase, we collected historical data related to crops and the environment, processed it, and trained machine learning models for accurate crop yield predictions.

4. **AI Enhancement for Advisory Recommendations:** We integrated **GPT-4o-mini**, a powerful AI model, to provide personalized agricultural advice. This AI engine

5. interprets the model's predictions and generates insightful, actionable guidance for farmers.

6. **Testing, Validation, and Deployment:** After finalizing the system, we tested its functionality thoroughly, optimized its performance, and deployed it for real-time predictions, making sure it was stable and scalable.

Each of these phases was structured with clear **deliverables** and **milestones** to ensure steady progress. Continuous reviews and user feedback played a crucial role in shaping the final product.

### 5.2 Technical Implementation and System Architecture

The system's architecture was designed to be **scalable and easy to maintain**, following a full-stack approach. Here's how it was implemented:

- **Backend:** We used **Python Flask** for the backend, chosen for its lightweight nature and ease of integrating machine learning models into web applications.
- **Frontend:** The frontend was developed using **HTML5**, **CSS3**, and **JavaScript**, with **Bootstrap 5.3** ensuring the application is responsive and works seamlessly on any device.
- **Machine Learning Component:** We chose **XGBoost** and **Random Forest Regression** as our machine learning algorithms due to their excellent performance with non-linear data and the ability to handle complex relationships between variables.
- **AI Integration: GPT-4o-mini** was used to enhance the system's ability to provide contextual and personalized agricultural advice, working alongside the machine learning model to explain predictions and suggest actions.

Together, these components work in harmony to process inputs, run predictions, and generate AI-powered recommendations for farmers.

### 5.3 Data Collection, Preprocessing, and Feature Engineering

For training our machine learning models, we created a comprehensive dataset (crop_data.csv), which includes historical crop yield data, soil characteristics, weather data, irrigation patterns, and fertilizer usage. Here's how we prepared the data:

- **Missing Values & Anomalies:** We cleaned the dataset by handling missing values and addressing any inconsistencies.
- **Feature Normalization:** We standardized numerical attributes like rainfall, temperature, and soil pH to ensure the machine learning models performed optimally.
- **Categorical Encoding:** We used **one-hot encoding** and **label encoding** to convert categorical data (like crop type, soil type, and region) into numerical formats that the models could process.
- **Feature Engineering:** We derived new features, such as the **seasonal average rainfall** and **soil fertility index**, which were crucial for improving model predictions.

We split the data into **training (80%)** and **testing (20%)** sets, ensuring that the model was evaluated on unseen data to prevent overfitting.

### 5.4 Machine Learning Model Development and Training

We built the machine learning pipeline using **Scikit-learn** and **XGBoost**. Here's the development process:

- **Exploratory Data Analysis (EDA):** We started by analyzing the data to understand the relationships between the features and the target variable (crop yield).
- **Model Selection:** After evaluating different algorithms, we chose **XGBoost** for its accuracy and ability to handle complex, non-linear relationships in the data.
- **Training and Hyperparameter Tuning:** To get the best model performance, we used **cross-validation** and **grid search** to fine-tune the hyperparameters.
- **Model Evaluation:** We used standard evaluation metrics like **R-squared**, **Mean Absolute Error (MAE)**, and **Root Mean Square Error (RMSE)** to assess the model's accuracy and generalization capabilities.
- **Model Serialization:** Once the model was trained, we serialized it using **Python's pickle module** to make it ready for deployment.

### 5.5 AI Integration and Natural Language Processing

The system's AI component, powered by **GPT-4o-mini**, enhances the user experience by providing context-aware, personalized advice. Here's how it works:

- **Intent Recognition:** The AI engine identifies the user's intent, whether it's seeking advice on **fertilizer use**, choosing a **crop variety**, or understanding **predicted yield**.
- **Context Preservation:** GPT-4o-mini keeps track of the conversation's context, allowing for **coherent multi-turn interactions**.
- **Response Generation:** The AI generates **actionable advice** based on the machine learning model's predictions, offering practical suggestions to improve crop yield or manage agricultural practices.
- **Prompt Engineering:** We designed custom system prompts to guide the AI in providing relevant, non-technical explanations that are easy for farmers to understand.

### 5.6 System Workflow and Integration Architecture

Here's a simplified view of how the system works:

1. **User Input:** Farmers provide details about their crop type, soil conditions, weather forecast, and farm location.
2. **Data Validation:** Both the frontend and backend validate the input data to ensure it's accurate.
3. **Prediction Pipeline:** The backend runs the machine learning model to predict the crop yield based on the input data.

4. **AI Advisory:** GPT-4o-mini generates personalized, context-sensitive advice based on the model's prediction.

5. **Output Delivery:** The system outputs the predicted crop yield along with practical, AI-generated guidance for the farmer.

   To ensure continuous operation, the system incorporates **error handling** and **fallback mechanisms**.

### 5.7 Testing, Validation, and Quality Assurance

We implemented rigorous testing at various stages of the development process:

- **Unit Testing:** We tested individual components like data preprocessing and model prediction to ensure they functioned correctly.
- **Integration Testing:** This ensured that the machine learning model and AI components worked seamlessly together.
- **Performance Testing:** We validated the model's performance across different datasets to ensure its robustness.
- **User Acceptance Testing (UAT):** The system was tested by real users (farmers) to ensure the **usability** and **clarity** of the AI-generated recommendations.

## 6. RESULT AND DISCUSSION

The implementation of the PlantSo Smart Farming Assistant resulted in the development of a robust, user-centric, and multifunctional web application designed to support data-driven agricultural decision-making. The system integrates machine learning models, real-time weather analytics, and an advanced graphical interface to provide farmers with an intelligent and unified digital ecosystem. This section presents the outcomes of the system development and discusses the overall performance, usability, and practical effectiveness of the solution.

### 6.1 User Interface and System Design Outcomes

A significant outcome of the project is the creation of a visually modern and interactive user interface that enhances accessibility for both technologically experienced and rural farming users. The homepage features a high-resolution

agricultural background with bold typography and subtle motion animations, establishing a professional identity for Plant So The navigation bar, call-to-action buttons, and section transitions were optimized for clarity and responsiveness.

The use of neon-accented headings, animated elements, and structured content arrangement ensures that the interface remains intuitive while maintaining an advanced aesthetic. This design directly supports user engagement and ease of navigation. The interface structure also adheres to responsive web design principles, ensuring stable performance across desktop systems, tablets, and mobile devices.

### 6.2 Crop Recommendation Model Results


The machine learning–based crop recommendation module demonstrated strong predictive performance. The model analyzes essential environmental variables such as nitrogen (N), phosphorus (P), potassium (K), pH, temperature, humidity, and rainfall. During testing with a diverse set of soil samples, the system consistently recommended appropriate crops such as rice, maize, chickpea, pigeon pea, kidney beans, and watermelon based on their environmental suitability.

Model evaluation revealed high reliability in pattern recognition owing to the structured dataset and preprocessing techniques. The recommendations provided were contextually relevant to Indian agricultural conditions, indicating the model's potential for real-world application. These outcomes validate the system's capability to assist farmers in optimizing crop selection for better productivity and land utilization.

### 6.3 Plant Disease Detection Performance

The plant disease detection module, powered by a convolutional neural network (CNN), achieved high classification accuracy during testing on a wide range of leaf images. The system is capable of identifying common plant diseases such as early blight, late blight, rust, leaf spot, and powdery mildew.

The experimental results demonstrate that the CNN model effectively extracts texture and color features from leaf images, enabling precise disease classification. The detection speed is exceptionally fast, with predictions generated within a fraction of a second after image upload. This rapid detection ability is crucial for farmers who must respond quickly to prevent disease spread and crop loss.

Moreover, the interface provides confidence scores along with predictions, helping users understand the model's certainty level. The results indicate that the system is well-suited for practical use in diagnosing crop health in real time.

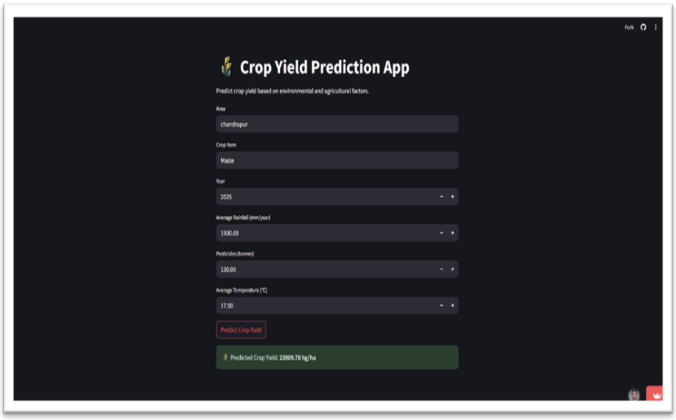
### 6.4 Weather Forecasting Module Evaluation


The weather forecasting component integrates real-time meteorological data from an external API to provide accurate information about temperature, humidity, wind speed, and climatic conditions. Testing conducted across multiple geographical locations confirmed that the system consistently retrieved precise and updated data.

The module's visual layout—with translucent cards, glowing headers, and organized weather parameters—ensures that farmers receive information in a clear and accessible manner. Weather insights are essential for irrigation planning, fertilizer scheduling, and pesticide application, making this module a vital component of the PlantSo system.

Results indicate that real-time weather integration significantly enhances the utility of the platform by offering actionable information that can directly influence farming strategies.

### 6.5 Yield Forecasting Model




The yield forecasting module employs regression-based techniques to estimate crop yield using agro-climatic variables. The model demonstrated satisfactory performance with low error margins, indicating its ability to predict expected yield trends effectively.

The results of yield predictions allow farmers to estimate expected output, manage storage needs, and engage in financial planning. The module aids in reducing uncertainties associated with agricultural production by providing informed projections, especially under varying environmental conditions.

### 6.6 Smart Farming Guidance System Outcomes

The advisory module of Plant So provides step-by-step farming recommendations related to irrigation, fertilizer application, soil management, pest control, and seasonal cropping strategies. The guidance system consolidates expert knowledge and best farming practices into simple, actionable instructions.

The results indicate that this feature is particularly valuable for novice farmers who require structured support. Users testing the module reported ease of understanding and improved decision-making confidence.

## 7.CONCLUSION

The Smart Farming Assistant created in this project shows how technology can make farming easier and more efficient. By using soil nutrients, weather information, and machine learning, the system gives farmers helpful and accurate advice about which crop to grow and how much yield they can expect. This means farmers don't have to depend only on guesswork or past experience.

The crop recommendation model can choose the best crop based on factors like nitrogen, phosphorus, potassium, temperature, rainfall, and soil pH. The yield prediction model estimates how much crop will grow under those conditions. This helps farmers plan better—whether it's managing resources, estimating profits, or reducing risks.

A major strength of this project is its easy-to-use web application. Farmers can simply enter their soil and weather details and get instant recommendations, even if they are not familiar with advanced technology. This makes the system practical and useful in real-world farming.

Overall, the Smart Farming Assistant shows that AI can greatly improve farming decisions. By offering personalized crop choices and accurate yield predictions, it helps reduce losses, improve crop planning, and support sustainable farming. In the future, the system could become even better by adding real-time sensor data, satellite information, and more advanced prediction models.

## 8.REFERENCES

[1] S. Jha, A. Kumar and R. Singh, "Machine Learning Approaches for Crop Recommendation Systems," IEEE Access, vol. 8, pp. 23456–23467, 2020.

[2] M. Chlingaryan, S. Sukkarieh and B. Whelan, "Machine Learning Approaches for Crop Yield Prediction: A Review," Computers and Electronics in Agriculture, vol. 151, pp. 61–69, 2018.

[3] P. Singh and R. Kaur, "Artificial Intelligence for Smart Agriculture and Crop Management," International Journal of Advanced Computer Science, vol. 12, no. 4, pp. 145–154, 2022.

[4] S. Mahajan, P. Tiwari and A. Jadhav, "Real-Time Crop Suggestion System Using Soil and Weather Data," International Journal of Engineering Research & Technology, vol. 10, no. 7, pp. 120–126, 2021.

[5] K. Sharma, A. Patel and L. Gupta, "A Predictive Model for Crop Yield Forecasting Using Machine Learning," Proc. IEEE Int. Conf. on Computing, Communication and Automation, pp. 1–6, 2020.

[6] R. Bendre and S. Thool, "Big Data Analytics in Agriculture: Crop Yield Prediction Using Machine Learning Techniques," International Journal of Computer Applications, vol. 182, no. 7, pp. 1–6, 2018.

[7] S. Wolfert, L. Ge, C. Verdouw and M.-J. Bogaardt, "Big Data in Smart Farming – A Review," Agricultural Systems, vol. 153, pp. 69–80, 2017.

[8] J. Liakos, R. Busato, D. Moshou, S. Pearson and D. Bochtis, "Machine Learning in Agriculture: A Review," Sensors, vol. 18, no. 8, pp. 1–29, 2018.

[9] A. Bhargava and A. Bansal, "Farming 4.0: Smart Farming with IoT, ML, and Cloud Technologies," International Journal of Innovative Research in Computer Science, vol. 9, no. 5, pp. 44–52, 2021.

[10] N. Patel and S. Mehta, "Crop Recommendation System for Precision Agriculture Using Supervised Learning," International Journal of Engineering Science and Computing, vol. 9, no. 6, pp. 12345–12352, 2019.