Indian Institute of Information Technology, Design and Manufacturing,Kurnool

Project Report

Smart Agriculture Assistant: An AI-Powered Decision Support System

Student Name: Harshit Verma (122CS0023), Rikhuo Vese (122CS0037), Shivam Kataria (122CS0038)

Project Guide: Dr. N. Srinivas Naik

Date of Submission: 15/09/2025

1. Introduction

Modern agriculture operates at the intersection of complex variables, including climate change, soil health, pestilence, and economic market forces. For farmers, making optimal decisions in such a dynamic environment is a significant challenge. While Artificial Intelligence (AI) and Machine Learning (ML) offer powerful tools to analyze these variables, they often exist as isolated, academic models that are not integrated into a practical platform for the end-user. This project addresses this gap by developing a "Smart Agriculture Assistant," a holistic, web-based decision support system designed to empower farmers with actionable, AI-powered insights. The application successfully integrates three distinct machine learning models and several data-driven features to provide a comprehensive suite of tools, including crop recommendation, real-time plant disease detection, weed identification, a fertilizer calculator, and market price information.

2. Literature Review

A review of current literature reveals a strong focus on applying AI to specific, isolated agricultural tasks. Numerous studies demonstrate the efficacy of Convolutional Neural Networks (CNNs) for image-based classification of plant diseases and weeds, often achieving high accuracy on curated datasets. Similarly, traditional machine learning models like Random Forest have been successfully used for crop yield prediction and crop selection based on historical weather and soil data. The reference paper for this project, "XAI-Powered Smart Agriculture Framework for Enhancing Food Productivity and Sustainability," highlights a key trend: the need for integrated systems that are not just accurate but also explainable and trustworthy. It proposes a conceptual framework where multiple data sources feed into a unified system, a concept which heavily guided the architecture of our project.

3. Research Gaps Identified

The literature review identified several key gaps that this project aimed to address:

1. **Lack of Integration:** Most tools solve only one problem (e.g., just disease detection). There is a need for a single, unified platform where a farmer can access multiple AI-driven advisories.
2. **Accessibility Gap:** Many powerful AI models remain within research papers and are not deployed in easy-to-use applications for non-technical users.
3. **Data Fusion Challenge:** Combining diverse data types—such as tabular climate data, soil properties, and image data—into a cohesive and functional system presents a significant technical challenge that is often overlooked in specialized research.

4. Problem Statement

To design, develop, and deploy an integrated web application that leverages multiple machine learning models to provide farmers with actionable, data-driven recommendations for crop selection, disease and weed identification, and awareness of market prices and fertilizer needs, thereby enhancing productivity and sustainability.

5. Objectives

1. To gather and preprocess diverse datasets: historical crop yield, gridded weather, and large-scale image collections.
2. To develop a **Random Forest Classifier** to recommend the most suitable crop based on a user's specific environmental and soil conditions.
3. To develop two **Convolutional Neural Network (CNN)** models for the real-time identification of plant diseases and common weeds from user-uploaded images.
4. To integrate these three models into a single, user-friendly web application using a Python/Flask back-end and an HTML/JavaScript front-end.
5. To enhance the application with practical supplementary features, such as a fertilizer calculator and a market price tracker.

6. Proposed Methodology

The project followed a structured, multi-stage methodology.

- **Data Gathering:** Data was sourced from multiple reputable platforms: historical district-level crop area and production data from ICRISAT; satellite-based daily weather data from the NASA POWER project API; and two large, labeled image datasets for plant diseases (PlantVillage) and weeds (V2 Plant Seedlings) from Kaggle. Market price data was also sourced from a Kaggle dataset derived from Indian government sources.
- **Data Preprocessing:** This was the most complex phase. The primary challenge was fusing the district-level crop data with the gridded weather data. This was solved by implementing a **nearest-neighbor spatial merge**. First, district center coordinates were obtained using geocoding via the geopy library. Then, for each district-year combination, the geographically closest weather data point was identified and merged using a cKDTree algorithm for efficiency. This process created a single, unified master dataset (final_cleaned_data.csv) for training the recommendation model.

- **Model Development:**
    1. **Crop Recommender:** A RandomForestClassifier was trained on the final merged dataset. The model was trained to predict the Crop category based on input features including state, season, and various weather and soil parameters. To improve real-world utility, the training data was filtered to focus only on major, non-ambiguous crops. The final model was enhanced to provide a ranked list of the top three most suitable crops with confidence scores.
    2. **Image Models (CNNs):** Two separate Convolutional Neural Network models were built and trained using TensorFlow and Keras in a Google Colab environment, leveraging free GPU resources for efficiency. The models were trained to a high level of accuracy on their respective datasets.

7. Proposed Architecture

The application was built using a standard client-server architecture.

- **Backend (Server-Side):** A Python-based web server using the **Flask** micro-framework. This server is responsible for loading the three pre-trained AI models, providing API endpoints for each feature, processing user input, and returning model predictions as JSON.
- **Frontend (Client-Side):** A single-page application built with standard **HTML**, styled with **Tailwind CSS**, and made interactive with **JavaScript**. The front-end uses the fetch API to communicate with the Flask back-end without needing to reload the page.

8. Implementation and Results

The project resulted in a fully functional, multi-tabbed web application.

- **Image Detection:** The disease and weed detection tabs allow users to upload an image and receive an instant classification. A confidence threshold was implemented to filter out irrelevant or unclear images, improving user experience.
- **Crop Recommendation:** The recommendation tab features a comprehensive form where users can input their field's conditions. The back-end model processes this data and returns a ranked list of the top three crop recommendations.
- **Additional Features:** The application was successfully enhanced with a fertilizer calculator and a market price tracker that uses a reliable local CSV file. The UI also includes a "Fetch Live Weather" button that uses the browser's geolocation to auto-fill weather data from a live weather API.

9. Conclusion / Final Results

This project successfully demonstrates the feasibility of creating an integrated, AI-powered decision support system for agriculture. By combining a data-driven crop recommender with real-time diagnostic tools and economic data, the "Smart Agriculture Assistant" provides a powerful, holistic platform that directly addresses the needs of farmers. The successful fusion of diverse data types and the deployment of multiple models into a single, user-friendly interface represents a significant technical achievement. Future work could focus on deploying the application to a public cloud service and further refining the models with more granular data.

10. References

A. Datasets:

1. **Crop Production Data:**
    - *Source:* International Crops Research Institute for the Semi-Arid Tropics (ICRISAT).
    - *Dataset:* District Level Database (DLD) for India.
    - *Link:* http://data.icrisat.org/dld/
2. **Weather Data:**
    - *Source:* NASA Langley Research Center (LaRC) POWER Project.
    - *Dataset:* POWER (Prediction of Worldwide Energy Resources) Daily Gridded Data.
    - *Link:* https://power.larc.nasa.gov/data-access-viewer/
3. **Plant Disease Image Data:**
    - *Source:* Kaggle Datasets.
    - *Dataset Name:* New Plant Diseases Dataset (Augmented).
    - *Creator:* VIPIN KUMAR.
    - *Link:* https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset
4. **Weed & Seedling Image Data:**
    - *Source:* Kaggle Datasets.
    - *Dataset Name:* V2 Plant Seedlings Dataset.
    - *Creator:* The Aarhus University Department of Engineering (ENG).
    - *Link:* https://www.kaggle.com/datasets/vbookshelf/v2-plant-seedlings-dataset
5. **Market Price Data:**
    - *Source:* Kaggle Datasets (Originally from data.gov.in).
    - *Dataset Name:* Agricultural Product Prices in India.
    - *Creator:* SURAJ B.
    - *Link:* https://www.kaggle.com/datasets/suraj520/agricultural-product-prices-in-india

B. Key Technologies & Libraries:

1. **Python:** The core programming language for the back-end and model development.
2. **Flask:** A micro web framework for Python, used to build the web server and API.
3. **Pandas:** A library for data manipulation and analysis, used extensively for cleaning and merging the datasets.
4. **Scikit-learn:** A machine learning library for Python, used to build the Random Forest Classifier for crop recommendation.
5. **TensorFlow & Keras:** Deep learning frameworks used to build, train, and deploy the Convolutional Neural Network models for image classification.
6. **Geopy & SciPy:** Libraries used for geocoding district names to coordinates and performing the nearest-neighbor spatial merge.