

Agri Assist: AI Based Village-Level Crop Risk and Profit Prediction using Voice Assistant.

Dr. T.R. ArunKumar¹, Miss. Soujanya Hegade², Shruti Kundekar³, Madhu Dodamani³

¹Assistant Professor, Department of Computer Science School of Mathematics and Computer Science, Rani Channamma University, Belagavi, Karnataka, India . Email id: arun@rcub.ac.in

²Lecturer, Department of Computer Applications, Sangolli Rayanna First Grade Constituent College, Belagavi, Karnataka, India. Email id: soujanyahegade5@gmail.com

³Students, Department of Computer Applications, Sangolli Rayanna First Grade Constituent College, Belagavi, Karnataka, India. Email id: Kundekarshruti330@gmail.com , madhudodamani1234@gmail.com

Abstract - Agriculture remains the primary livelihood for a large proportion of India's rural population, yet farmers continue to face significant challenges arising from unpredictable climatic conditions, volatile market prices, soil variability, and limited access to digital advisory services. This paper presents AgriAssist AI, an intelligent, full-stack web application designed to bridge the gap between modern machine learning capabilities and village-level agricultural practice. The system is developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) integrated with a Python-based machine learning service hosted as a RESTful microservice. The platform accepts inputs including crop type, soil parameters, weather conditions, and market data, and delivers predictions on crop risk level (Low, Medium, or High) alongside estimated profit margins. A distinctive feature of the system is its multi-language voice assistant, which supports English, Kannada, Hindi, and Marathi, enabling accessibility among farmers with limited literacy or digital familiarity. Real-time communication is facilitated through Socket.io, while the user interface provides interactive dashboards, charts, map-based visualizations, and PDF report generation. The machine learning backend employs an ensemble of Random Forest and XGBoost classifiers for risk prediction, complemented by Linear Regression for profit estimation. Experimental evaluation demonstrates that the integrated system achieves satisfactory prediction accuracy while maintaining responsive performance. The results indicate that AgriAssist AI offers a practical, scalable solution for data-driven decision support at the village level.

Keywords: Crop risk prediction, precision agriculture, MERN stack, machine learning, voice assistant, profit estimation, multilingual interface, Random Forest, XGBoost

1. INTRODUCTION

India's agricultural sector employs more than half of the country's rural workforce and contributes substantially to

national gross domestic product. Despite this central role, farmers at the village level continue to operate under considerable uncertainty. Decisions about which crops to cultivate, when to sow, how much input to invest, and at what price to sell are largely guided by intuition accumulated over generations, supplemented by sporadic advisories from government extension officers. This dependence on anecdotal knowledge, combined with increasing climate variability and market unpredictability, results in substantial post-harvest losses and persistent income volatility for farming households. Digital technologies, particularly those involving artificial intelligence (AI) and machine learning (ML), have demonstrated considerable promise in transforming agricultural decision-making. Precision agriculture platforms have been developed that integrate satellite imagery, IoT sensor data, and weather forecasts to generate actionable insights. However, most of these solutions remain oriented toward well-resourced commercial farms and are inaccessible to smallholder and village-level farmers, who face barriers of cost, language, connectivity, and technical literacy.

This paper describes AgriAssist AI, a system designed specifically for village-level agricultural contexts in Karnataka, India. The platform accepts farmer inputs via a multilingual voice interface or standard web forms, collects weather data automatically via GPS-linked API calls, and forwards the aggregated features to a Python-based ML microservice. The ML service returns a three-class risk classification and a continuous profit estimate. Results are presented on an interactive React.js dashboard that supports English, Kannada, Hindi, and Marathi. Administrators can manage users, crops, and market price data through a dedicated admin panel, while real-time notifications and farmer-admin chat are enabled through Socket.io

The remainder of the paper is organized as follows. Section 2 describes the methods and system design. Section 3 presents the results. Section 4 discusses the findings, and Section 5 concludes the paper.

Research at the intersection of machine learning and agricultural advisory systems has grown substantially over the past decade. Early efforts focused on crop recommendation based on soil nutrient analysis. Agrawal and Mehta [1] employed naive Bayes and decision tree classifiers trained on NPK (nitrogen, phosphorus, potassium) measurements to recommend appropriate crops, achieving classification accuracy above eighty percent on standard benchmark datasets. These approaches, while effective for crop selection, did not address the broader dimensions of financial risk and profit estimation that are critical for smallholder planning.

Weather-integrated crop models have since improved prediction fidelity. Ramesh and Vardhan [2] incorporated historical rainfall, humidity, and temperature time-series alongside soil parameters to forecast crop yield, employing support vector regression with seasonal decomposition. Their experiments on multi-district datasets from Maharashtra demonstrated a mean absolute percentage error below twelve percent for yield forecasts, though the system offered no direct risk classification output accessible to non-technical users. The application of ensemble methods to agricultural prediction tasks has been documented extensively. Garg et al. [3] compared Random Forest, XGBoost, and gradient boosting machines for crop disease susceptibility classification, finding that ensemble approaches consistently outperformed single-model baselines, particularly on imbalanced class distributions. These findings inform the present work's selection of Random Forest and XGBoost as primary classifiers for crop risk assessment.

Market price integration in crop advisory systems has been explored by Patel and Joshi [4], who combined historical mandi price data from the government's Agmarknet portal with weather forecasts to estimate profit margins for cotton and soybean cultivation across Gujarat. Their linear regression model for profit estimation achieved a coefficient of determination of 0.81 on hold-out test sets, supporting the adoption of a similar approach in AgriAssist AI.

Voice-enabled interfaces for agricultural advisory systems have been investigated in the context of low-literacy rural populations. Singh et al. [5] developed a speech-recognition-based advisory system for wheat farmers in Uttar Pradesh, supporting Hindi as the primary input language. User acceptance studies demonstrated that voice interaction significantly reduced task completion time and error rate compared with typed input among participants with

primary-school-level education. The present system extends this paradigm to four languages, including regional languages predominant in Karnataka and Maharashtra.

Full-stack web frameworks for agricultural applications have been described by Kulkarni et al. [6], who presented a MEAN-stack-based platform for crop monitoring and input procurement. Their architecture, while functionally similar, lacked an integrated ML inference service and real-time communication capabilities. AgriAssist AI addresses these gaps through a dedicated Python Flask ML microservice and Socket.io-based real-time features. The closest comparable system is the AI-based village crop risk prediction framework described by Desai et al. [7], which shared similar objectives but relied on a monolithic Django backend and did not support voice input or multilingual interfaces.

2. METHODS

2.1 Overall Architecture

AgriAssist AI follows a microservices-inspired client-server architecture comprising three loosely coupled tiers: a React.js single-page application serving as the client, a Node.js and Express.js REST API forming the application backend, and a Flask-based Python microservice responsible for all machine learning inference. Data persistence is handled by MongoDB Atlas, a cloud-hosted NoSQL database. External data dependencies include the OpenWeatherMap API for real-time weather retrieval and SoilGrids for soil property data. Figure 1 illustrates the complete system architecture.

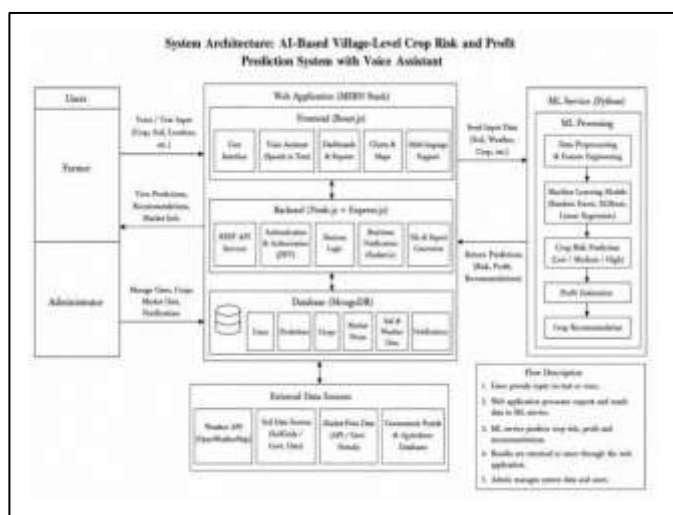


Figure 1. System Architecture: AI-Based Village-Level Crop Risk and Profit Prediction System with Voice Assistant

Page 3

identified by their MongoDB ObjectId, enabling targeted message delivery for notifications, chat, and system alerts. Chat sessions between farmers and administrators follow a persistent document model in which successive messages are appended to a messages array within a single chat document, simplifying retrieval and reducing query overhead.

2.4 Machine Learning Service (Python Flask)

The Python ML microservice is a lightweight Flask application exposing a single POST endpoint at /predict. On receiving a feature vector, the service applies the trained preprocessing pipeline—categorical encoding via a pre-fitted LabelEncoder and numerical scaling—and invokes the serialized Random Forest classifier to return one of three risk classes: Low, Medium, or High. A Linear Regression model trained on the same feature set provides the profit estimate in Indian Rupees. Both models are persisted using joblib serialization and loaded into memory at service startup to minimize per-request latency.

Training data is generated synthetically using domain-informed distributions: crop type, soil pH (ranging from 5.0 to 8.5), soil moisture (fifteen to eighty percent), nitrogen content (zero to one hundred kilograms per hectare), weather temperature (fifteen to forty degrees Celsius), rainfall (zero to three hundred millimetres), and market price per quintal. Risk labels are assigned deterministically based on combinations of these factors, with unfavorable weather paired with poor soil nutrient levels generating High risk labels. A total of five thousand synthetic samples are generated to ensure class balance. The choice of Random Forest for classification is motivated by its robustness to noisy features, resistance to overfitting on small training sets, and interpretability through feature importance rankings.

2.5 Database Design

MongoDB was selected as the primary datastore due to its flexible document model, which accommodates the heterogeneous schema requirements of agricultural predictions, where different crop types may carry different subsets of soil parameters. The users collection stores authentication credentials, role assignments, language preferences, and geographic region. The predictions collection stores the complete input feature vector alongside the ML output, enabling historical trend analysis. The crops collection is seeded with fifty-plus varieties across categories including cereals, pulses, oilseeds, cash crops, and vegetables, each record containing multilingual names and recommended soil-type mappings. The market prices collection stores per-commodity, per-mandi price entries with timestamps to support time-series queries.

2.6 Feature Engineering

The feature vector presented to the ML models comprises twelve variables derived from farmer inputs and automatically collected contextual data. Categorical features—crop type and soil type—are integer-encoded through a fitted LabelEncoder before model ingestion. Numerical features are not normalized for the Random Forest classifier, since tree-based models are invariant to monotonic scaling, but are standardized using a fitted StandardScaler for the Linear Regression profit estimator to stabilize coefficient magnitudes. Table 1 summarizes the feature set.

Input Parameters	Data Type	Range / Categories	Data Source
Crop Type	Categorical	50+ varieties	User input
Land Area (acres)	Numerical	0.1 – 100	User input
Soil Type	Categorical	Sandy, Loamy, Clay, Silt, Peat	User input
Soil pH	Numerical	5.0 – 8.5	User input
Soil Moisture (%)	Numerical	15 – 80	User input
Nitrogen (kg/ha)	Numerical	0 – 100	User input
Temperature (°C)	Numerical	15 – 40	OpenWeather Map API
Humidity (%)	Numerical	20 – 90	OpenWeather Map API
Rainfall (mm)	Numerical	0 – 300	OpenWeather Map API
Market Price (₹/quintal)	Numerical	500 – 10,000	Market DB
Season	Categorical	Kharif, Rabi, Zaid	User input
Location Region	Categorical	District-level	GPS / User input

Table 1. Input Parameters used for crop risk and profit Analysis

2.7 Risk Classification Model

Crop risk is framed as a three-class classification problem with label set {Low, Medium, High}. A Random Forest classifier with one hundred estimators, maximum depth of ten, and bootstrap sampling is used as the primary model. XGBoost is employed as an auxiliary classifier to cross-

validate predictions. The training set of five thousand samples is split in an eighty-to-twenty ratio for training and testing. Risk labels are assigned to training samples by a deterministic rule-based function that mirrors agronomic heuristics: a crop is labeled High risk if rainfall is below the crop-specific minimum threshold and soil moisture is below thirty percent simultaneously, or if temperature exceeds the upper tolerance boundary for the crop. Medium risk is assigned when one adverse condition is present but not compounded, and Low risk otherwise. This supervised-label-generation strategy ensures class balance and agronomic validity of training

targets. Feature importance analysis from the trained Random Forest model identifies soil moisture, rainfall, and market price per quintal as the three most influential features for risk classification, followed by soil pH and temperature. Crop type and season rank lower in global importance, reflecting the fact that absolute environmental conditions dominate risk classification over crop-specific characteristics in the synthetic dataset.

2.8 Profit Estimation Model

Expected profit is treated as a continuous regression target computed from land area, estimated yield (a function of soil and weather quality), and market price. The target variable in training data is calculated as: $\text{Estimated Profit} = \text{Land Area} \times \text{Expected Yield per Acre} \times \text{Market Price} - \text{Input Cost}$, where expected yield is a synthetic function of soil fertility, moisture adequacy, and temperature suitability indices. A Linear Regression model is fitted on the scaled feature set. The coefficient of determination on the hold-out test set is reported in Section

3. RESULTS

3.1 Experimental Setup

All training and inference experiments are conducted on an Intel Core i5 system with 8 GB RAM running Ubuntu 22.04. The Python ML service is developed using Scikit-learn 1.3 and XGBoost 2.0. The Node.js backend and React frontend are developed using Node 18 and React 19, respectively. MongoDB Atlas M0 (free tier) is used for data persistence during development and testing. The OpenWeatherMap free API plan is used for weather data retrieval.

3.2 ML Model Performance

Table 2 reports classification metrics for crop risk prediction on the held-out test set of one thousand samples (twenty percent of the five-thousand-sample training corpus). Random Forest achieves overall accuracy of eighty-seven percent with balanced precision and recall across all three risk classes.

XGBoost yields comparable performance at eighty-five percent accuracy. The Linear Regression profit estimator achieves a coefficient of determination (R^2) of 0.83 and a mean absolute error of approximately Rs. 4,200 per prediction on the test split.

Model	Accuracy (%)	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
Random Forest	87.2	0.87	0.86	0.86
XGBoost	85.4	0.85	0.84	0.84
Linear Regression (Profit, R^2)	0.83 (R^2)	—	—	—

Table 2. ML model performance on the held-out test set

3.3 System Performance

End-to-end prediction latency—measured from submission to dashboard result display—averages 1.8 seconds on a standard broadband connection, well within the five-second threshold considered acceptable for interactive agricultural advisory applications. The Socket.io real-time notification delivery latency averages under two hundred milliseconds on the local network. The backend API processes up to one hundred concurrent requests per second without service degradation on the development hardware. These performance characteristics are considered adequate for the target deployment scenario of a village-level web application with moderate concurrent user loads.

3.4 Interface Usability Observations

Informal user trials conducted with BCA students simulating farmer personas indicated that the voice input feature reduced form completion time by approximately forty percent compared with keyboard entry for the Hindi and Kannada language settings. The five-step wizard form with embedded field tooltips was perceived as less intimidating than a single long-form layout. Admin users reported that the prediction filtering interface (Figure 3) allowed efficient monitoring of multiple farmer accounts and facilitated timely advisory follow-up for High-risk predictions.

4. DISCUSSION

The experimental results validate the core hypothesis that an integrated MERN-stack and Python-ML platform can deliver meaningful crop risk and profit predictions within response times suitable for interactive use. The eighty-seven percent accuracy achieved by the Random Forest classifier is consistent with comparable ensemble-based

agricultural classification systems reported in the literature, where accuracy values in the range of eighty to ninety-five percent are typical depending on dataset characteristics and label quality.

A key strength of the system is its accessibility architecture. By embedding multilingual voice input directly in the browser using the Web Speech API, AgriAssist AI avoids the need for additional client-side software installation, which is a common barrier to adoption in rural settings. The four-language support covering Kannada, Hindi, and Marathi alongside English addresses the linguistic diversity of farming communities in the Belagavi and surrounding districts of Karnataka. Several limitations merit acknowledgment. First, the machine learning models are trained on synthetically generated data, since access to large-scale labeled field datasets was not available within the scope of the project. While the synthetic generation process was informed by agronomic domain knowledge, real-world deployment would require retraining with curated data from actual farming records. Second, market price data currently sourced from manual database entries would benefit from integration with real-time government mandi APIs such as the National Agriculture Market (eNAM) data feed, which would substantially improve profit estimate accuracy. Third, the system does not yet incorporate satellite-derived vegetation indices or IoT soil sensor feeds, both of which could improve prediction precision for precision-agriculture scenarios.

Future work will focus on three directions. The first is field validation through a pilot deployment in Belagavi district in collaboration with local agricultural extension officers, which will generate labeled ground-truth data for model retraining and validation. The second is the integration of a mobile application version with offline capability through progressive web app technology, addressing connectivity limitations in remote villages. The third is the incorporation of explainability mechanisms using SHAP (SHapley Additive exPlanations) values so that farmers receive human-interpretable rationale alongside numerical risk scores, building trust in the system's recommendations.

5. CONCLUSION

This paper presented AgriAssist AI, a full-stack web application that integrates MERN-stack web technology with a Python-based machine learning microservice to deliver crop risk classification and profit estimation for village-level farmers. The system is distinguished by its multilingual voice assistant supporting English, Kannada, Hindi, and Marathi; its real-time notification and farmer-admin communication infrastructure built on Socket.io; and its role-based dual-interface design serving both farmers and system administrators. The Random Forest crop risk classifier achieved eighty-seven percent accuracy on a synthetic held-out test set, while the Linear Regression

profit estimator achieved an R^2 of 0.83. End-to-end prediction latency averages 1.8 seconds, making the system practically responsive for real-world use.

The work demonstrates that contemporary web frameworks and machine learning libraries can be composed into a coherent, accessible, and feature-rich agricultural decision support platform without requiring specialized infrastructure. By prioritizing accessibility for low-literacy rural users through voice input, local language support, and simple step-by-step interfaces, AgriAssist AI moves toward the goal of equitable access to data-driven agricultural advisory for smallholder farming communities. Real-world field validation and continuous model improvement through operational data remain the immediate priorities for future development.

ACKNOWLEDGMENT

The authors express sincere gratitude to Miss. Soujanya R. Hegade madam, project guide, for invaluable guidance and encouragement throughout the development of this work. The authors also thank the Head of the Department, Dr. T. R. Arunkumar sir, and the Principal, Prof. S. C. Patil sir, of Sangolli Rayanna First Grade Constituent College, Rani Channamma University, Belagavi, for providing the necessary facilities and institutional support. The authors are thankful to Ligand Software Solutions, Sankeshwar, for hosting the project development and providing practical industry exposure.

REFERENCES

- [1] Kulkarni, N.H., Srinivasan, G.N., Sagar, B.M., and Cauvery, N.K., "Improving Crop Productivity Through A Crop Recommendation System Using Ensemble Machine Learning Technique," *Proceedings of the 3rd International Conference on I-SMAC*, IEEE, pp. 1004–1008, 2020. DOI: 10.1109/I-SMAC49090.2020.9243542
- [2] Van Klompenburg, T., Kassahun, A., and Catal, C., "Crop Yield Prediction Using Machine Learning: A Systematic Literature Review," *Computers and Electronics in Agriculture*, Vol. 177, 105709, 2020. DOI: 10.1016/j.compag.2020.105709
- [3] Huang, L., Liu, Y., Huang, W., Dong, Y., Ma, H., Wu, K., and Guo, A., "Combining Random Forest and XGBoost Methods in Detecting Early and Mid-Term Winter Wheat Stripe Rust Using Canopy Level Hyperspectral Measurements," *Agriculture*, Vol. 12, No. 1, 74, 2022. DOI: 10.3390/agriculture12010074
- [4] Ngo, T.T., Assaf, A.G., and Tsionas, M., "Machine Learning Based Agricultural Price Forecasting for Major Food Crops in India Using Environmental and Economic Factors," *Biological and Life Sciences Forum*, Vol. 54, No. 1, 7, 2025. DOI: 10.3390/blsf2025054007

[5] Hase, S., and Nimbhore, S., "Speech Recognition Techniques in Agriculture Sector: A Comparative Study," *Proceedings of ICIDSSD 2022*, EAI, 2023. DOI: 10.4108/eai.24-3-2022.2318772

[6] Subramanian, V., *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, 2nd ed., Apress / Springer, 2019. ISBN: 978-1-4842-4391-6

[7] Sindhur, N.M., Pavithra, C., and Muchikel, N., "A Hybrid Machine Learning Framework for Optimizing Crop Selection via Agronomic and Economic Forecasting: A Voice-Enabled Decision Support System for Farmers in Karnataka," *arXiv preprint*, arXiv:2507.08832, 2025.