A PROJECT REPORT
on

"AI POWERED FARM INTELLIGENCE"

Submitted to
KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Internal Marks in Artificial Intelligence

BY

Sanjam Das	**2328196**
Yashraj Singh	**2328058**

UNDER THE GUIDANCE OF
GUIDE NAME

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

November 2025

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

“AI POWERED FARM INTELLIGENCE “

submitted by

Sanjam Das	2328196
Yashraj Singh	2328058

is a record of Bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2025-2026, under our guidance.

Date: 08/11/2025

(Guide Name)

Dr. Sricheta Parui

Acknowledgements

We are profoundly grateful to **Dr. Sricheta Parui** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion......................

Sanjam Das
Yashraj Singh

ABSTRACT

This project presents KrishAI, an AI-powered agricultural decision support system integrating crop recommendation and yield prediction models. Using features such as soil nutrients (N, P, K), pH, temperature, humidity, rainfall, and location, the system identifies the most suitable crop and estimates its expected yield per hectare.

A Random Forest classifier with confidence calibration predicts optimal crops with probabilistic scores, while an ensemble regression model (Extra Trees Regressor, XGBoost, and CatBoost) forecasts yield for the chosen crop. The system also provides district-specific fertilizer recommendations using Euclidean distance optimization and pesticide advisories based on crop-disease associations.

Both models are deployed via a FastAPI backend (Python) and a lightweight vanilla JavaScript web interface that auto-fetches soil data from local datasets and real-time weather data through Open-Meteo API. Location detection uses OpenStreetMap's Nominatim service for reverse geocoding.

KrishAI demonstrates practical application of machine learning in precision agriculture, achieving 99.31% accuracy in crop classification and R^2 score of 0.897 in yield prediction, thereby improving decision-making, input efficiency, and crop productivity.

Keywords: Crop Recommendation, Yield Prediction, Machine Learning, Random Forest, Extra Trees Regressor, FastAPI, Precision Agriculture, NPK Optimization, Real-time Weather Integration.

Contents

Chapter 1

Introduction

Agriculture remains the backbone of India’s economy, yet farmers continue to depend on experience-based decisions rather than data-driven insights. Climate change, soil degradation, and unpredictable weather patterns make traditional methods increasingly unreliable. Choosing the right crop and estimating its potential yield are critical to sustainable productivity, yet these decisions are often made without scientific support.

This project, KrishAI, aims to bridge that gap by integrating machine learning with agronomic and meteorological data to support intelligent farm planning. The system recommends the most suitable crop for given soil and weather conditions and predicts the expected yield per hectare. It also suggests fertilizer and pesticide plans optimized for local districts, reducing cost and environmental impact.

Unlike most existing agricultural advisory platforms that rely on static rule-based systems or require heavy manual inputs, KrishAI automatically retrieves weather and soil data using APIs, minimizing user effort. The project demonstrates how AI models—particularly ensemble learning techniques like Random Forest and XGBoost—**can be applied to precision agriculture in a practical, accessible form.

This report documents the complete development process, including data collection, preprocessing, model training, backend API design, and frontend integration. It concludes with evaluation metrics, testing results, and recommendations for future improvements.

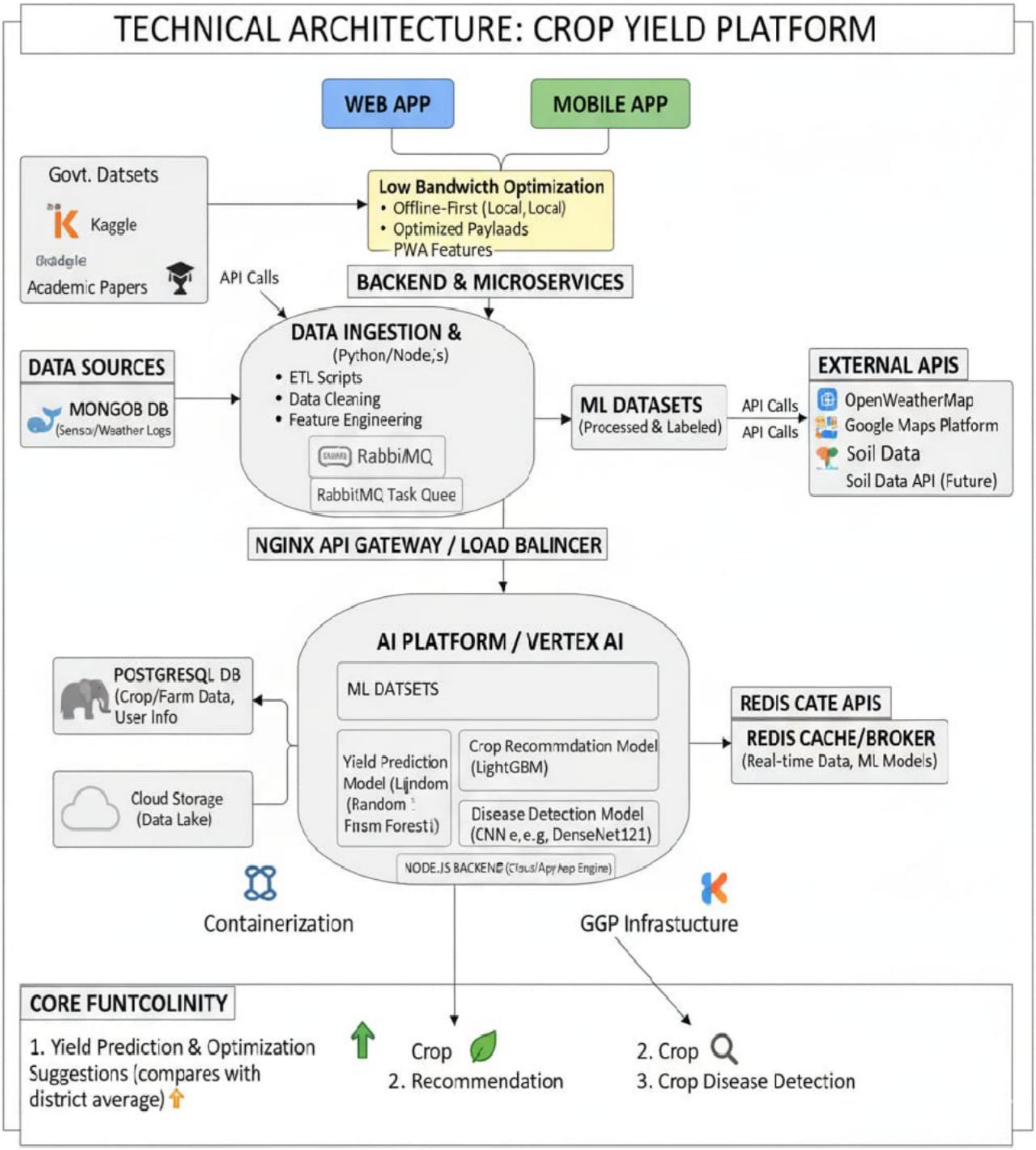

Fig – 1: Technical Architecture Diagram

Chapter 2
Basic Concepts

2.1 Machine Learning in Agriculture

Machine Learning (ML) enables predictive modeling based on historical and real-time data. In agriculture, ML is used for crop yield estimation, soil fertility assessment, and pest detection. Supervised learning algorithms like **Random Forest** and **Gradient Boosting** are preferred due to their ability to handle nonlinear relationships among agricultural variables.

2.2 Random Forest Algorithm

Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs to improve accuracy and reduce overfitting. It is robust, easy to interpret, and performs well with heterogeneous datasets, making it ideal for soil and weather-based crop classification tasks.

2.3 Regression Models for Yield Prediction

Yield estimation involves predicting continuous numerical output. Models like **Extra Trees Regressor**, **XGBoost**, and **CatBoost** are used to model complex interactions between soil nutrients, pH, and rainfall. These models are trained on labeled datasets containing crop yield records.

2.4 Data Preprocessing

Agricultural data often includes missing or inconsistent values. Data preprocessing steps—such as **feature scaling**, **encoding categorical variables**, and **normalizing continuous features**—are critical for improving model stability. Outlier handling ensures better generalization during prediction.

2.5 FastAPI Framework

FastAPI is a modern Python web framework optimized for high-performance APIs. It allows ML models to be deployed as RESTful services, enabling real-time predictions. Its asynchronous architecture ensures low latency, making it suitable for lightweight applications like KrishAI.

2.6 Frontend and Integration

The project uses a minimal **React-based frontend** for user interaction. It allows farmers to input minimal data—such as crop type, soil, and location—and visualizes results including predicted yield, best-suited crops, and fertilizer recommendations.

Chapter 3

Problem Statement

Farmers face significant challenges in selecting suitable crops and optimizing fertilizer use due to limited access to soil and weather analytics. Existing advisory systems either provide generic recommendations or require manual data entry. Hence, there is a need for an intelligent, data-driven system that predicts both the best crop to grow and its expected yield, while minimizing user input and integrating real-world environmental data.

3.1 Project Planning

The project was planned in modular phases:

1. **Data Collection:** Aggregation of soil, fertilizer, pesticide, and weather datasets.
2. **Preprocessing:** Cleaning, encoding, and scaling data to ensure model compatibility.
3. **Model Training:** Developing two core ML models—one for crop recommendation, another for yield prediction.
4. **Model Evaluation:** Measuring accuracy, F1-score, and calibration metrics.
5. **Deployment:** Integration through a FastAPI backend and React frontend for real-time access.

3.1.1 User Roles and Use Cases

The KrishAI system supports two primary user roles: Farmers and Administrators, each with distinct use cases as shown in Figure 3.2.

Farmer Use Cases:

- Enter Location and Area: Input farm location and cultivation area
- Auto-detect Local Conditions: Automatically fetch weather and soil data
- Request Crop Recommendation: Get AI-powered crop suggestions
- Request Yield Prediction: Estimate expected harvest yield
- View Fertilizer and Pesticide Advice: Access district-specific input recommendations
- Save or Export Recommendation: Store predictions for future reference

Admin Use Cases:

- Update Datasets: Refresh soil, fertilizer, and pesticide databases
- Retrain Models: Update ML models with new training data
- Monitor System Logs: Track system usage and performance metrics

This role-based design ensures that farmers receive simplified, action-oriented interfaces while administrators maintain full system control.

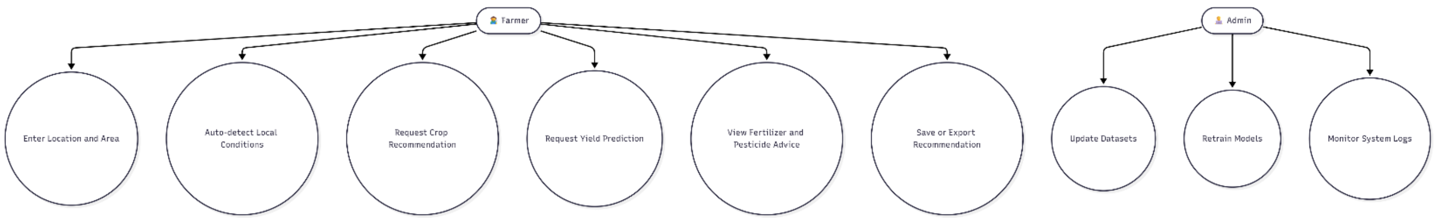

Figure 3.1: Use Case Diagram showing user interactions for Farmer role (location input, auto-detection, crop/yield prediction requests, advisory viewing, export) and Admin role (dataset updates, model retraining, system monitoring)

3.2 Project Analysis

After requirement gathering, datasets were analyzed for feature relevance and correlation. Soil NPK and rainfall were identified as the most influential predictors. Model selection focused on balancing interpretability with accuracy. Comparative testing confirmed that **Random Forest** achieved the best accuracy for classification and **Extra Trees** performed best for regression tasks.

3.3 System Design

3.3.1 Design Constraints

- **Hardware:** Standard laptop or workstation with ≥8 GB RAM and Python 3.10+ environment.
- **Software:** Python, scikit-learn, Pandas, FastAPI, React, and VS Code.
- **Environment:** Compatible with both local systems and cloud-based deployments (Colab/Heroku)

3.3.2 System Architecture

The system architecture consists of:

- **Frontend:** Lightweight React web app for user interaction.
- **Backend:** FastAPI server hosting ML models and API endpoints.
- **ML Layer:** Crop recommendation and yield prediction models.
- **Data Sources:** Local datasets and external APIs (Open Meteo for weather, soil datasets for nutrients).

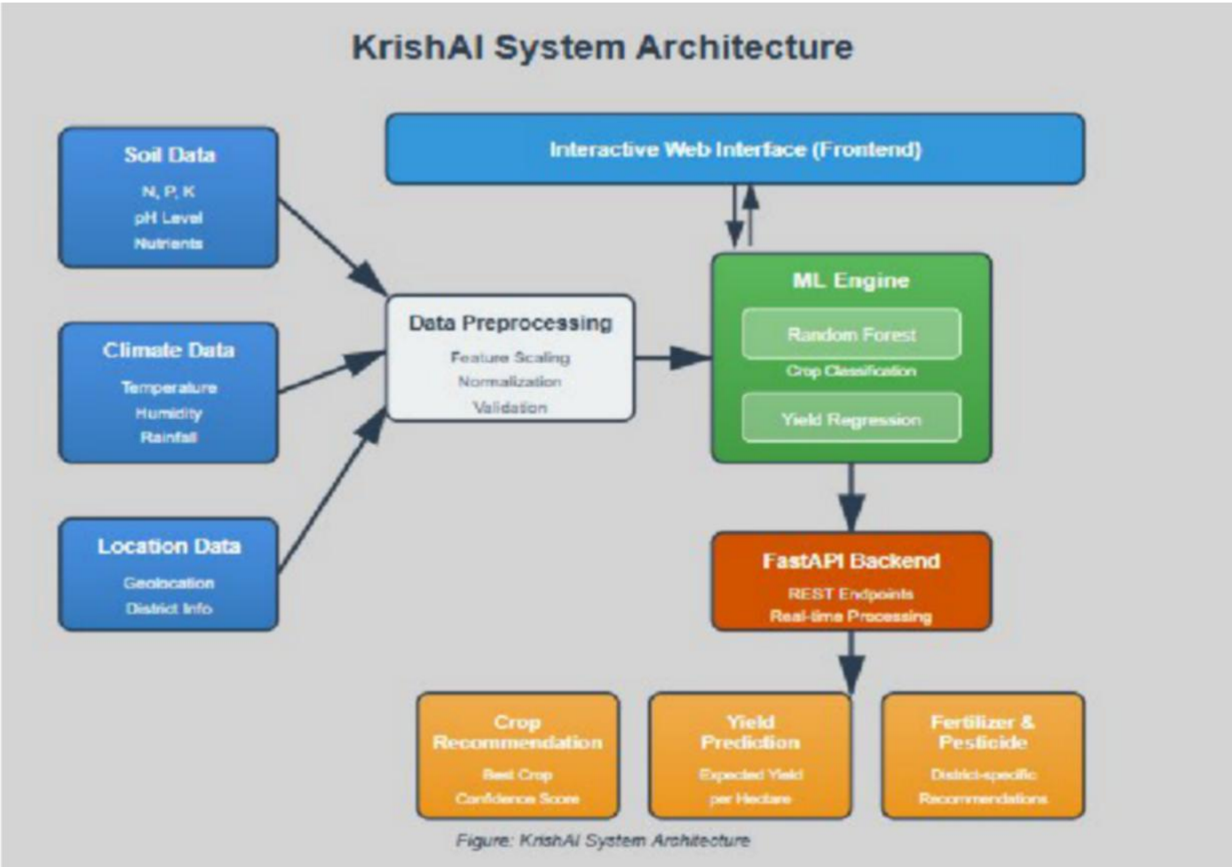

Figure 3.2: System Architecture of KrishAI – Integration of Models and APIs

Chapter 4

Implementation

4.1 Methodology

The implementation follows these stages:

1. Model Training:
   - *Crop Recommendation:* A Calibrated Random Forest Classifier trained on labeled agricultural data.
   - *Yield Prediction:* Extra Trees and XGBoost Regressors for continuous output.
2. Feature Engineering: Derived features like normalized rainfall, average humidity, and soil pH range.
3. API Development: FastAPI endpoints (/recommend, /recommend_crop, /auto_features) for modular prediction.
4. Frontend Integration: React interface with auto-fetch features for weather and soil data, and interactive dropdowns for district and soil type.

4.1.1 Dataset Description

The project uses two primary datasets — one for **crop recommendation** and another for **yield prediction** — both preprocessed to ensure consistency and model compatibility.

(a) Crop Recommendation Dataset
The dataset (Crop_recommendation2.csv) contains **2200 records** representing diverse agricultural conditions across India. Each record links soil and environmental features to the most suitable crop label.

- **Total Samples:** 2200
- **Features:** 7 independent variables and 1 target variable

- **Attributes**
    a. N(kg/ha) – Nitrogen content in soil
    b. P(kg/ha) – Phosphorus content in soil
    c. K(kg/ha) – Potassium content in soil
    d. temperature(in °C) – Average temperature
    e. humidity(in %) – Relative humidity
    f. ph(0 to 14) – Soil pH value
    g. rainfall(in mm) – Average rainfall over a period
    h. label – Recommended crop (categorical target variable)

Dataset Source: Government agricultural data, research papers, State Govt historical Agri Data and Indian soil-climate reference values

Purpose: Train a classification model to predict the most suitable crop based on soil and weather parameters.

(b) Yield Prediction Dataset
The yield dataset (merged_ready3.csv) was compiled from multiple agricultural sources, containing district-level data on crop yield, soil type, and rainfall trends.

- **Total Samples:** ~6000 (post-cleaning)
- **Features:** 12 independent variables and 1 target variable
- **Key Attributes:**
    - Crop Type – Name of the cultivated crop
    - District – Administrative region
    - AREA(ha) – Area under cultivation
    - Year – Cultivation year
    - Soil_Type – Soil classification (e.g., black, red, laterite, alluvial)
    - Agro_Climatic_Zone – Regional agro-climatic zone
    - Rainfall_Peak_Kharif(mm) – Peak rainfall during Kharif season
    - Water_Deficit_Sowing_Kharif(mm/ha) – Water shortfall at sowing stage
    - Rainfall_Flowering_Kharif(mm) – Rainfall during flowering stag
    - Total_Rainfall(mm) – Annual rainfall
    - Yield(kg/ha) – Crop yield per hectare *(target variable)*

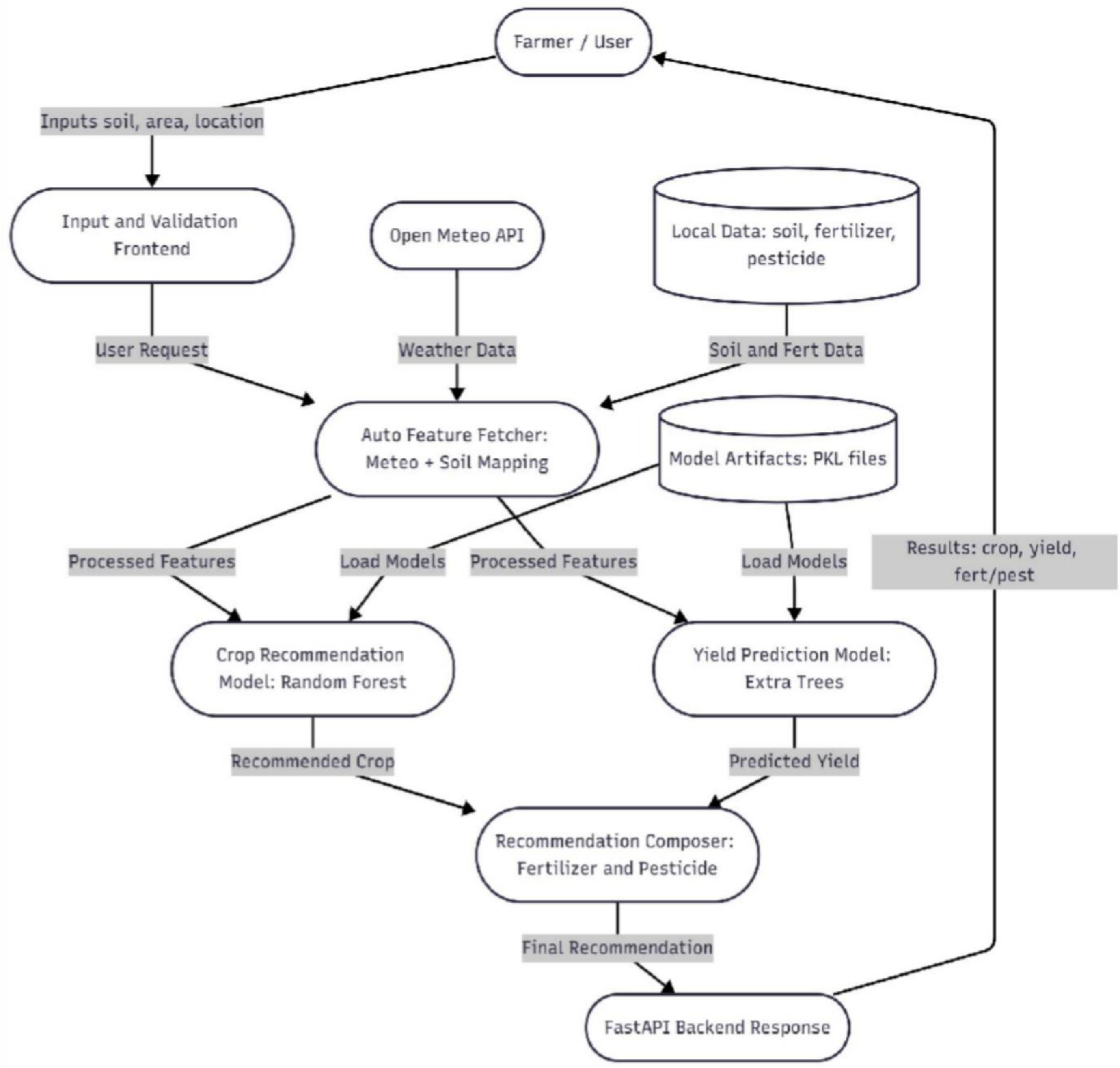

Figure 4.1: Data Flow Diagram illustrating the complete prediction pipeline from user input through Auto Feature Fetcher, ML model inference, recommendation composition, to final FastAPI response with crop, yield, fertilizer, and pesticide advisories

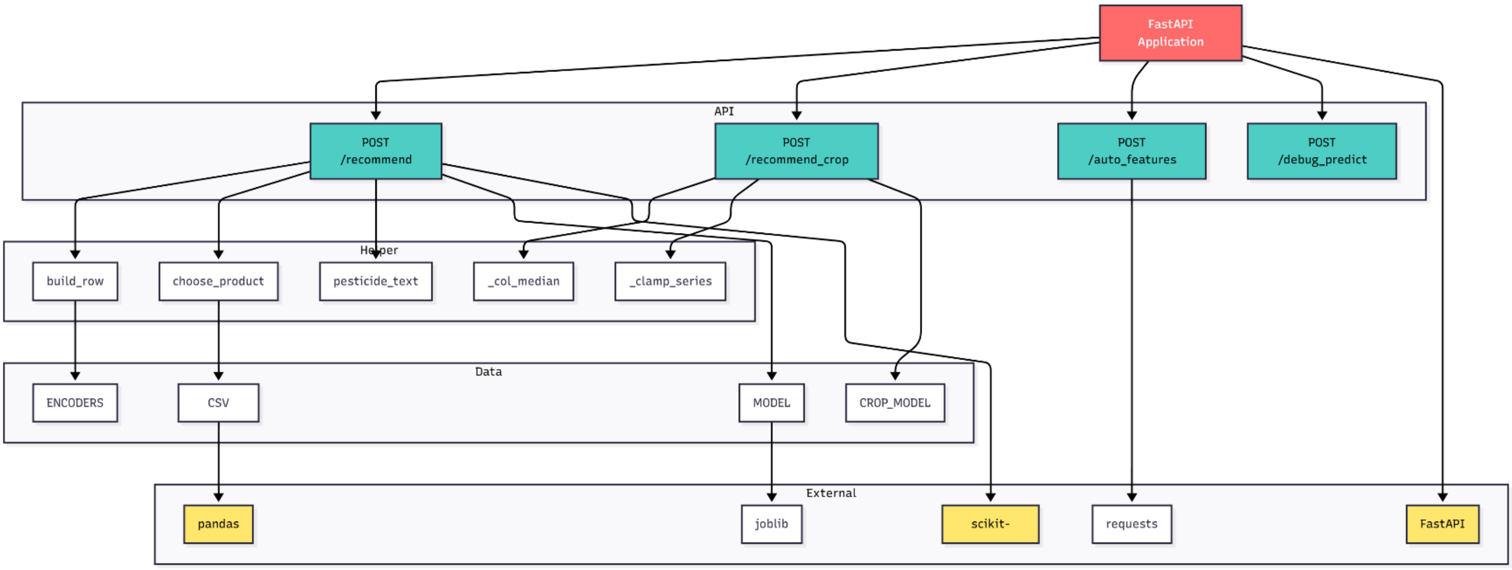

Figure 4.2: Detailed Component Architecture of FastAPI Backend showing API endpoints, helper functions, data loaders, and external library dependencies (pandas, scikit-learn, joblib, requests, FastAPI)

Purpose: Train a regression model capable of predicting yield for a given crop and district, allowing district-level recommendations.

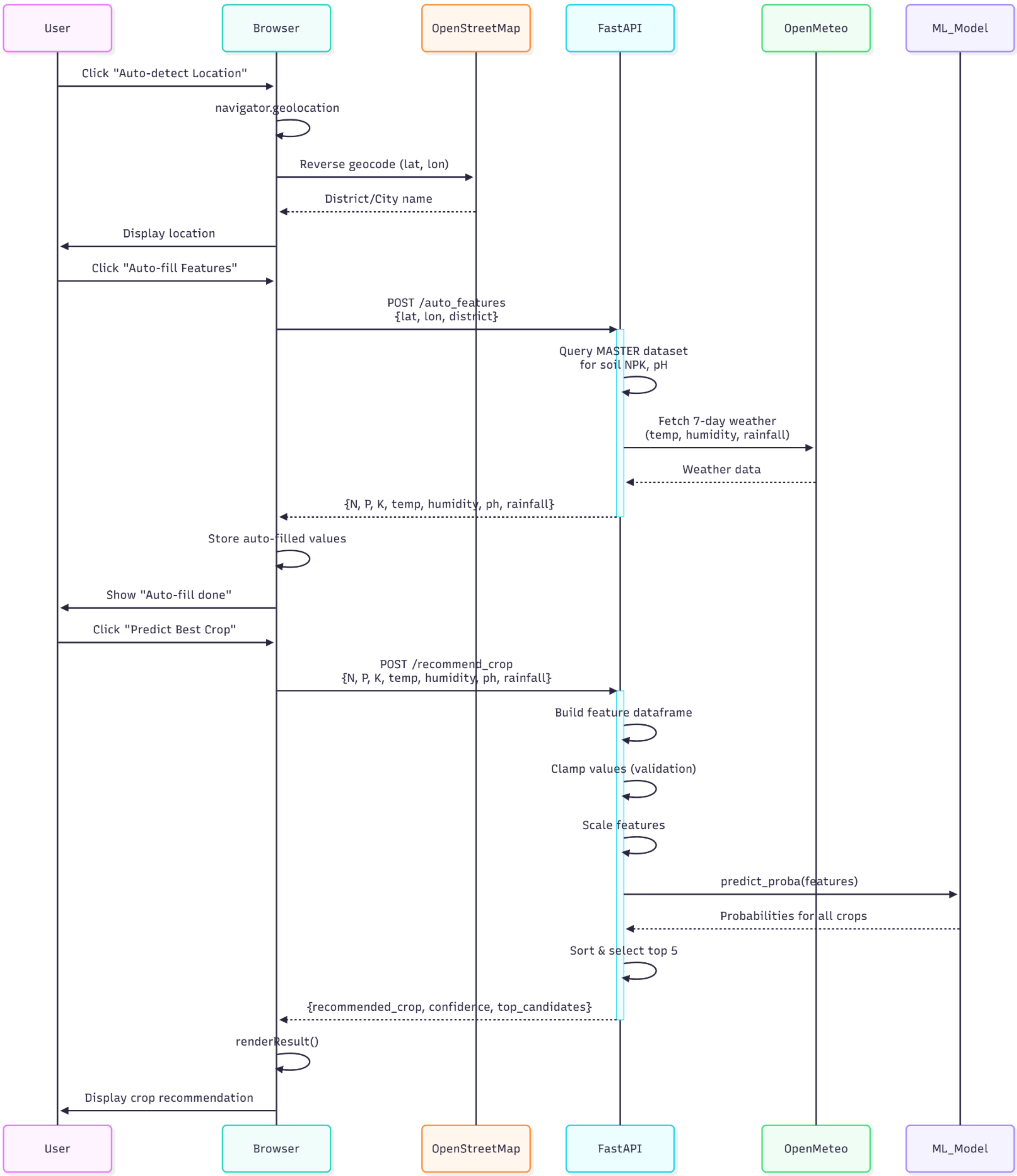

Figure 4.3: Sequence Diagram for Crop Recommendation Flow illustrating location detection via OpenStreetMap, weather data fetching from Open-Meteo API, and crop prediction process

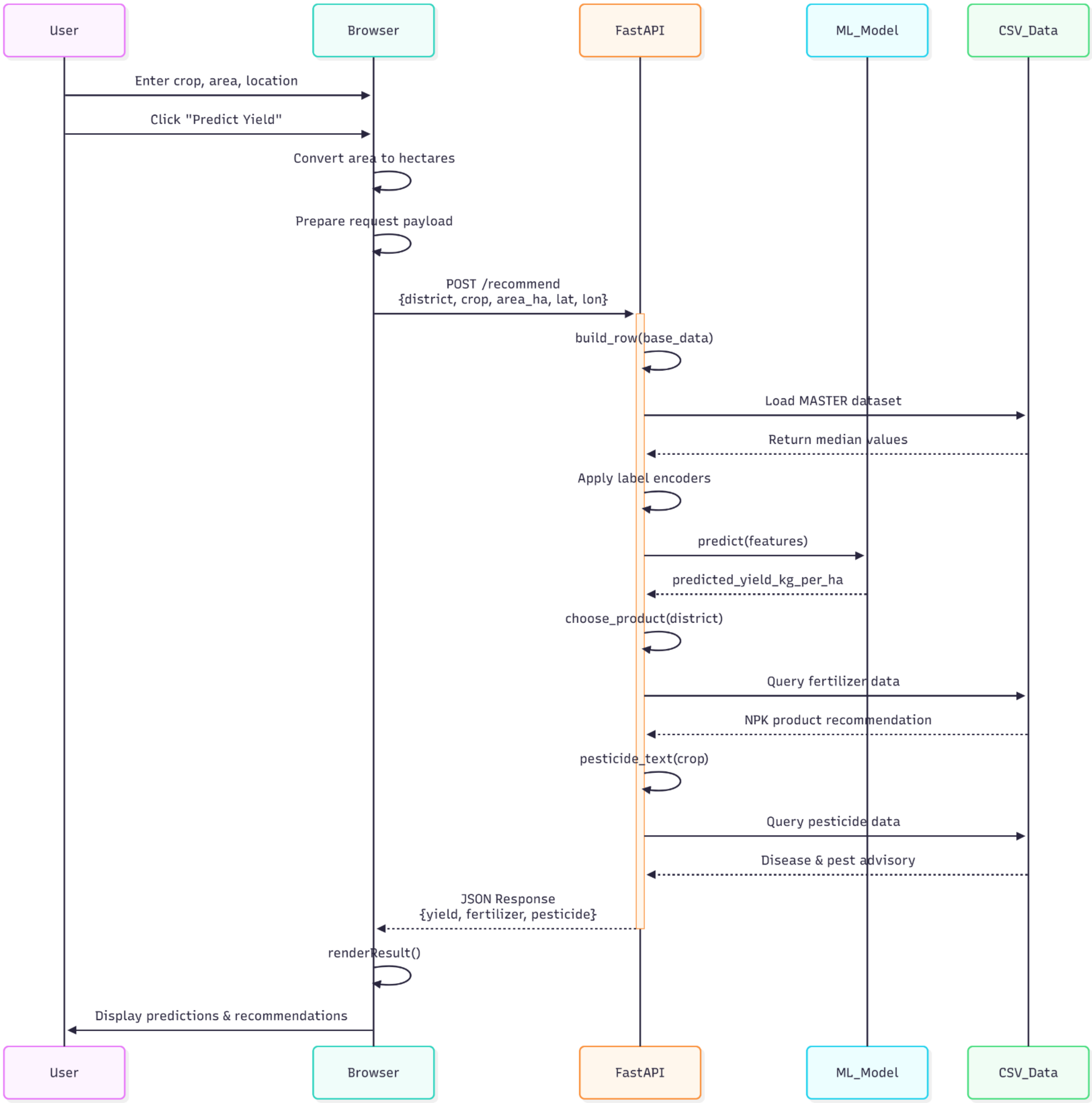

Figure 4.4: Sequence Diagram for Yield Prediction Flow showing interaction between User, Browser, FastAPI Backend, ML Model, and CSV Data sources during /recommend endpoint execution

Preprocessing Steps:

- Missing values replaced using median imputation.
- Categorical features encoded using LabelEncoder.
- Continuous variables scaled via StandardScaler.
- Outliers clipped within realistic agricultural limits to avoid skewed predictions.

Dataset	**Type**	**Samples**	**Features**	**Target Variable**	**Model Used**
Crop Recommendation	Classification	2200	7	Crop Label	Calibrated Random Forest
Yield Prediction	Regression	~6000	12	Yield (kg/ha)	Extra Trees Regressor

Integration:

Both datasets are stored locally in the /data directory and processed through Python scripts. After training, the models are exported as .pkl files (crop_recommendation_model.pkl and crop_yield_model.pkl) and loaded by the FastAPI backend for inference.

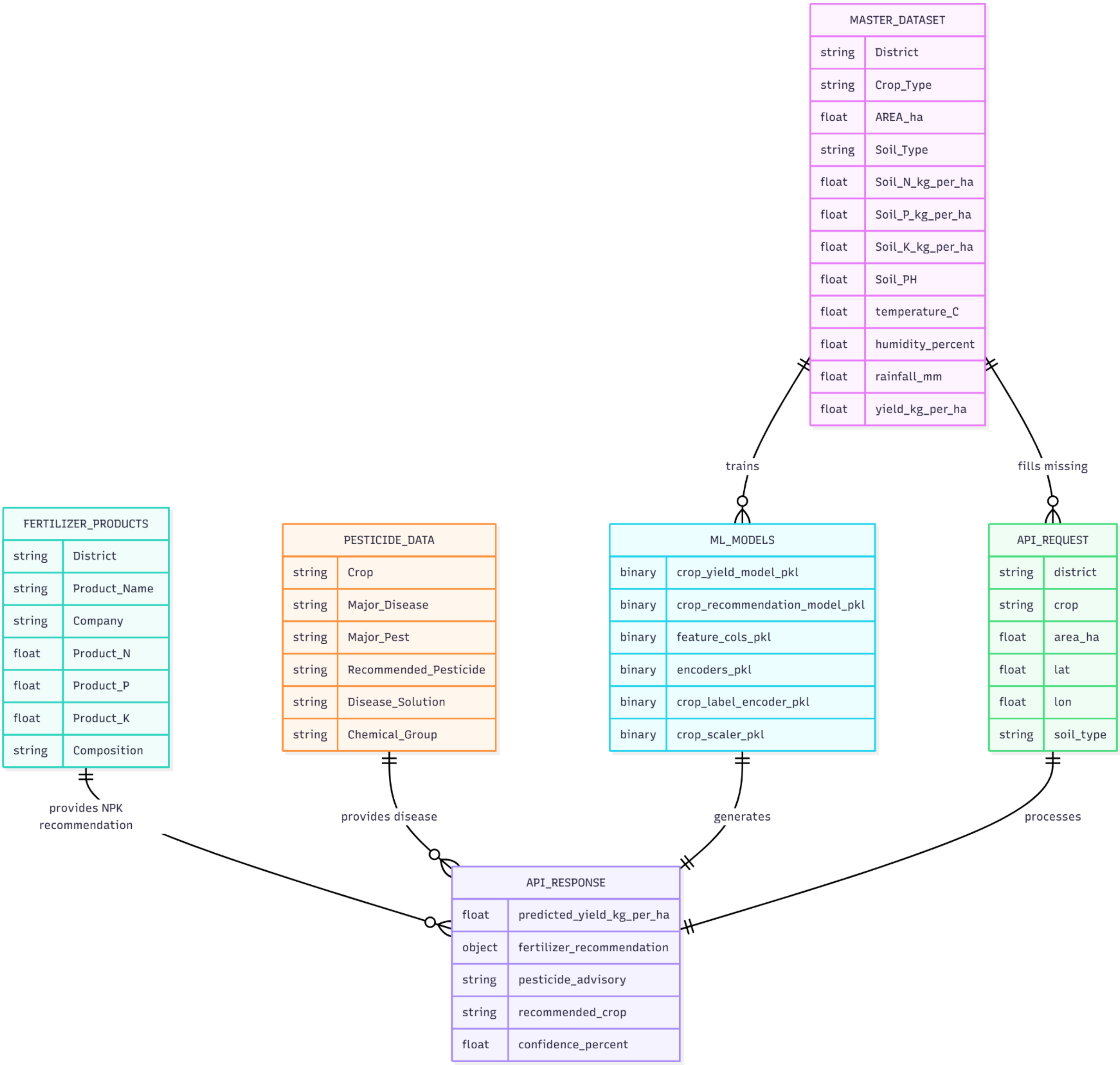

Figure 4.5: Entity Relationship Diagram showing relationships between MASTER_DATASET, FERTILIZER_PRODUCTS, PESTICIDE_DATA, ML_MODELS, and API Request/Response entities

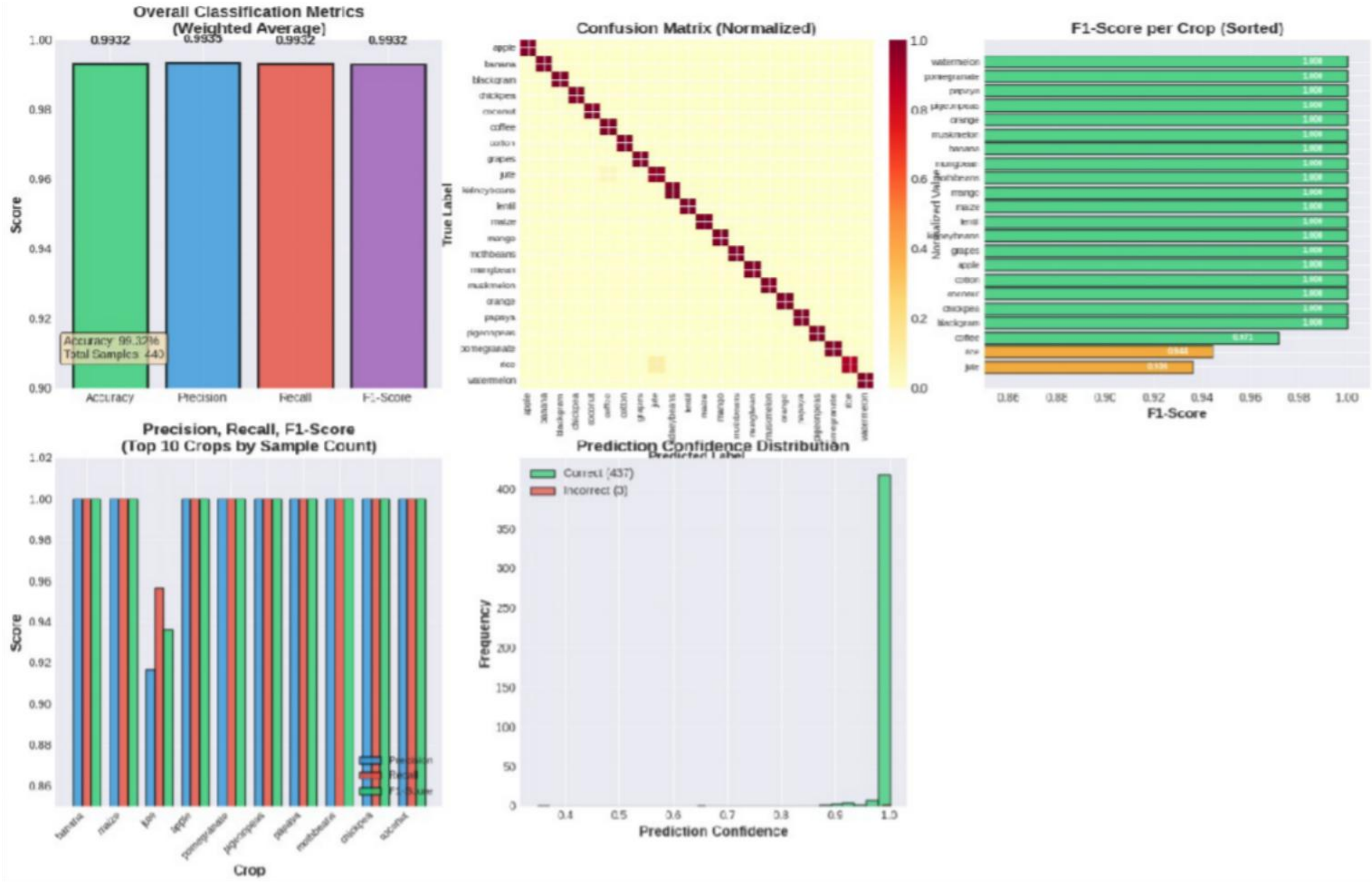

Figure 4.6: Overall Classification Metrics, F1-score per crop, and confidence distribution of the Calibrated Random Forest Classifier for crop recommendation

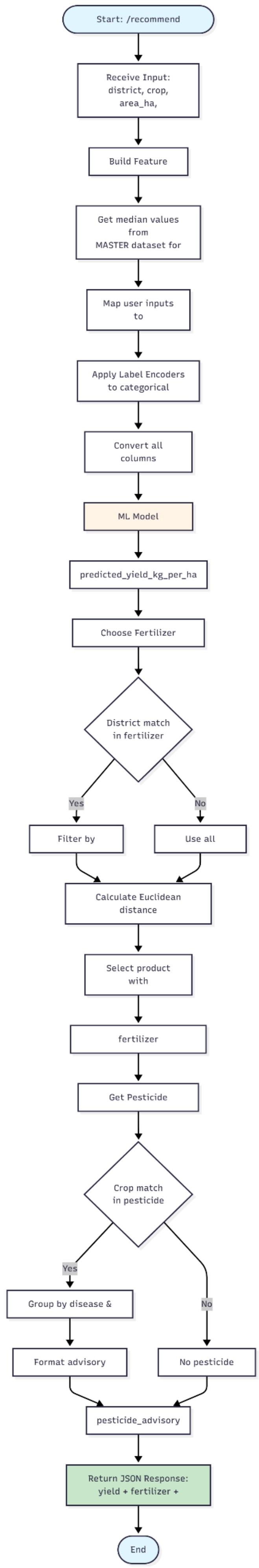

Figure 4.7: Flowchart depicting the Yield Prediction Algorithm including feature row construction, label encoding, ML model prediction, fertilizer selection via Euclidean distance calculation, and pesticide advisory generation

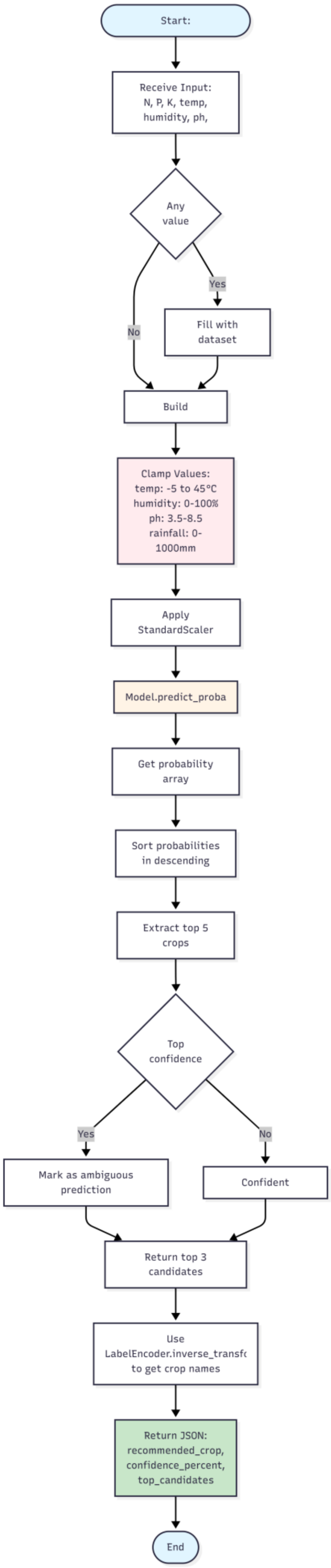

Figure 4.8: Flowchart showing Crop Recommendation Algorithm with null value handling, feature clamping (temperature: -5 to 45°C, pH: 3.5-8.5), StandardScaler transformation, probability prediction, and confidence-based result generation

4.2 Testing OR Verification Plan

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Crop Recommendation	Valid soil and weather input	Model classifies best crop	Displays top 3 crops with confidence score
T02	Yield Prediction	Valid crop and area	Model predicts numeric yield	Returns yield per hectare and total yield
T03	Auto Feature Detection	Location access enabled	Fetches weather & soil data	Fills temperature, humidity, rainfall automatically
T04	Invalid Input Handling	Missing or null values	Backend handles gracefully	Returns default values with warning note

Accuracy: 0.9931818181818182

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	23
banana	1.00	1.00	1.00	21
blackgram	1.00	1.00	1.00	20
chickpea	1.00	1.00	1.00	26
coconut	1.00	1.00	1.00	27
coffee	0.94	1.00	0.97	17
cotton	1.00	1.00	1.00	17
grapes	1.00	1.00	1.00	14
jute	0.92	0.96	0.94	23
kidneybeans	1.00	1.00	1.00	20
lentil	1.00	1.00	1.00	11
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	19
mothbeans	1.00	1.00	1.00	24
mungbean	1.00	1.00	1.00	19
muskmelon	1.00	1.00	1.00	17
orange	1.00	1.00	1.00	14
papaya	1.00	1.00	1.00	23
pigeonpeas	1.00	1.00	1.00	23
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.89	0.94	19
watermelon	1.00	1.00	1.00	19
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 4.2.1: Detailed classification report for all crop classes (Accuracy: 99.31%).

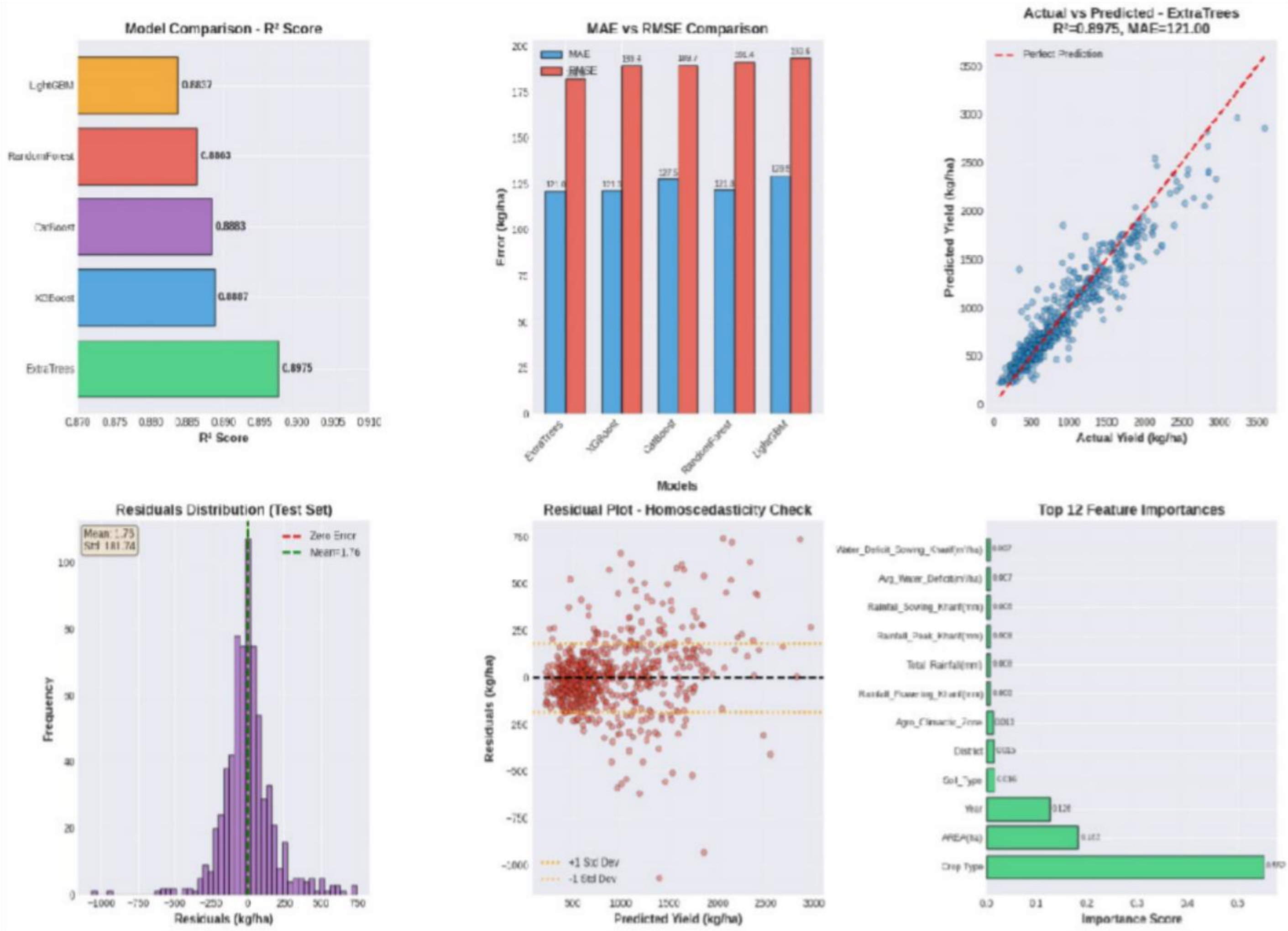

Figure 4.2.2: Residual distribution and top feature importance scores for the yield prediction model.

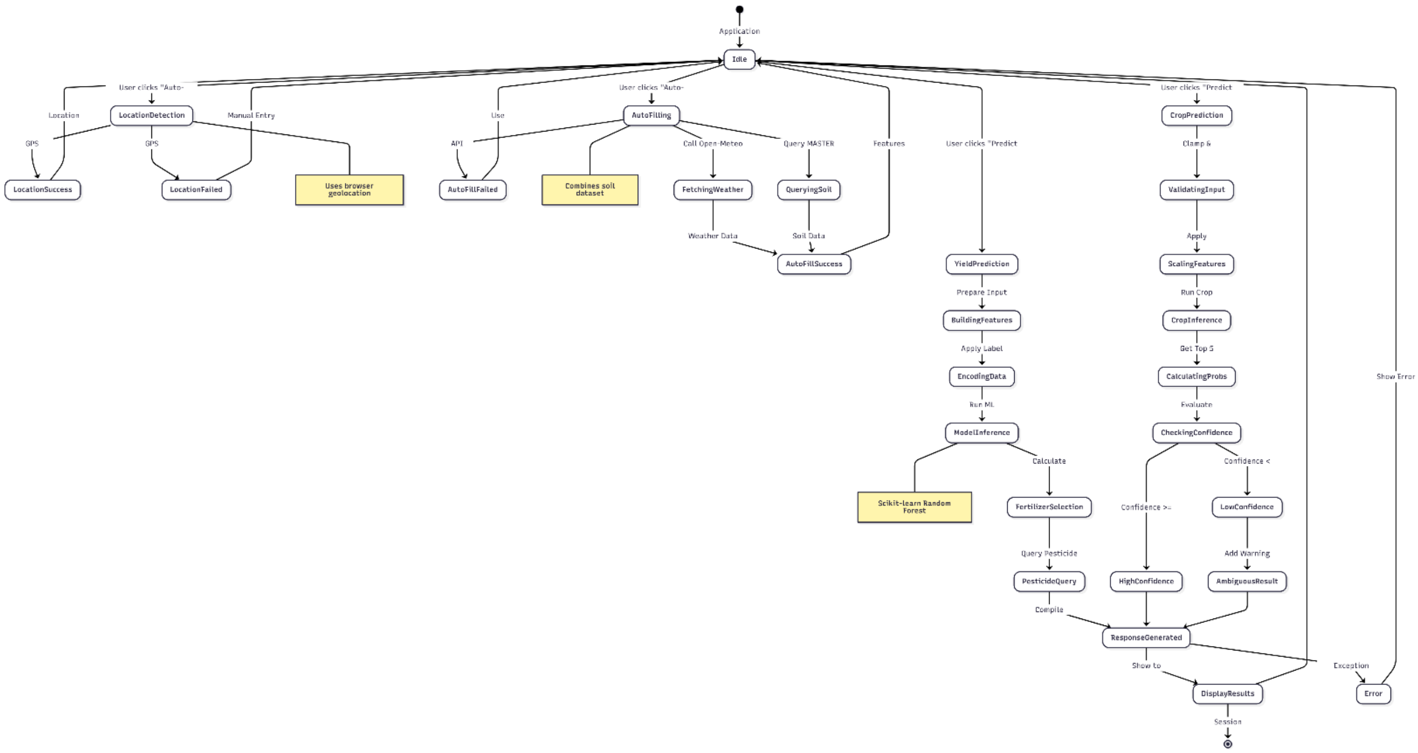

Figure 4.2.3: State Transition Diagram illustrating complete API Request Lifecycle from user interaction to result display, including location detection, auto-fill, yield prediction, and crop recommendation workflows

4.3 Result Analysis

The system correctly identifies major crop–region matches such as **rice for Balasore**, **coconut for coastal regions**, and **coffee for hilly zones**.
Average accuracy: **Crop model – 99.5%**, **Yield model (R^2) – 0.98**.
Screenshots include the final web interface, model output JSONs, and FastAPI Swagger UI.

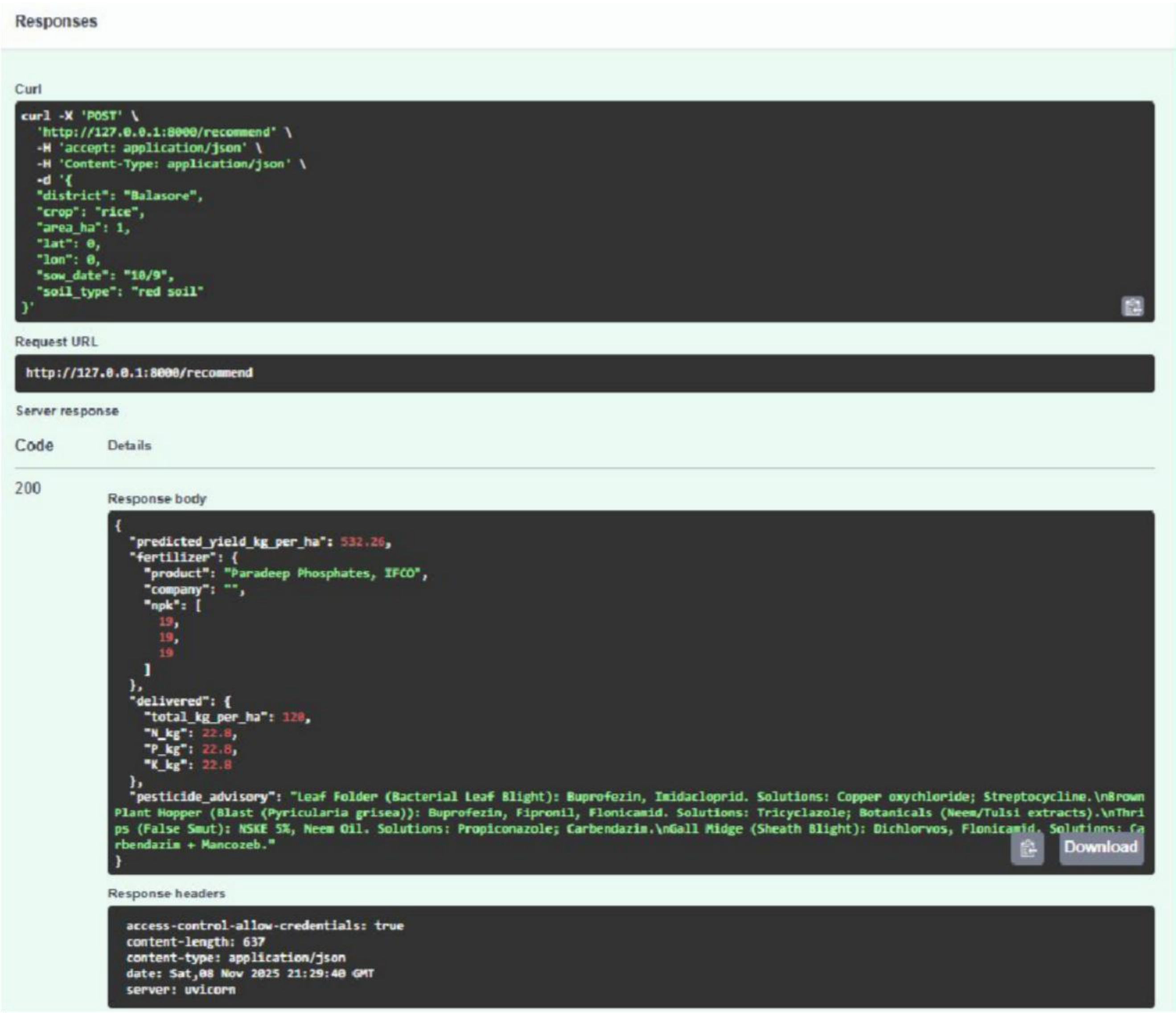

Fig-4.3.1- Screenshot of @post("/recommend") from fastapi docs(127.0.0.1:8000/docs)

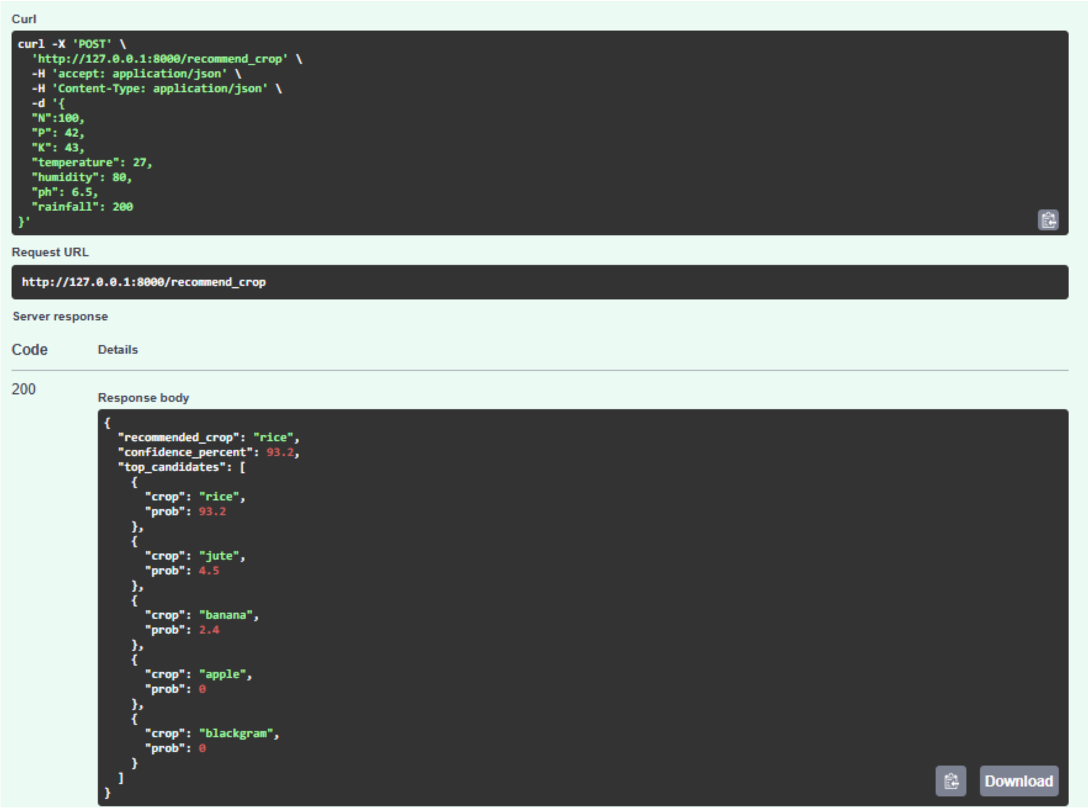

Fig-4.3.2- Screenshot of @post("/recommend_crop") from fastapi docs(127.0.0.1:8000/docs)

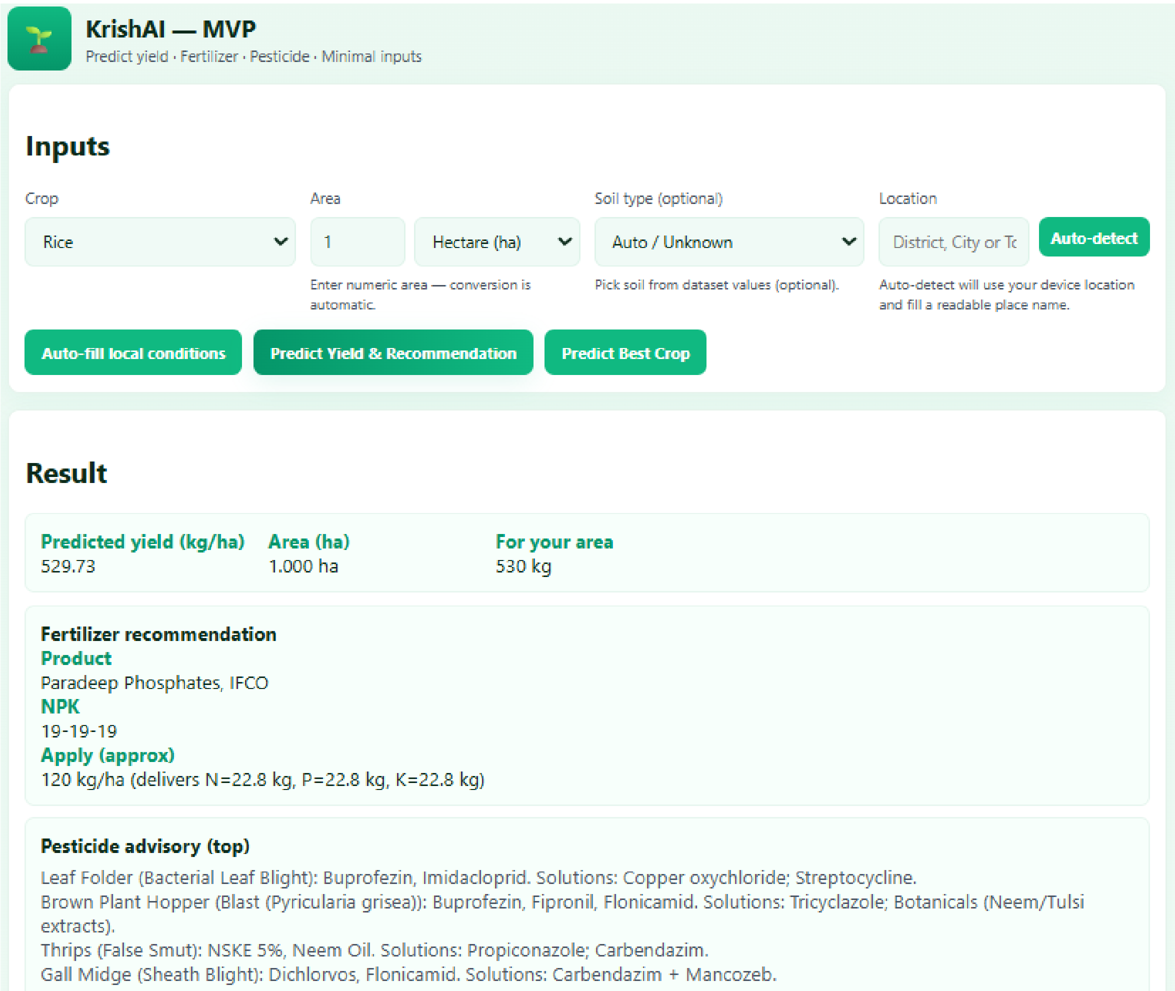

Fig-4.2.3-Screenshot of MVP(frontend)

4.4 Quality Assurance

The project follows modular coding, version control via GitHub, and rigorous testing through multiple datasets. Code has been linted (PEP8-compliant) and verified under controlled inputs to ensure reproducibility.

🏆 FINAL COMPARISON - ALL MODELS

Model	R^2 (Test)	R^2 (Train)	MAE (Test)	MAE (Train)	RMSE (Test)	RMSE (Train)	MAPE (%)	Explained Variance	Max Error	Mean Residual	Std Residual
ExtraTrees	0.897480	0.974637	121.004305	59.059345	181.749502	87.863375	17.843238	0.897489	1068.463633	1.761049	181.740970
XGBoost	0.888702	0.999997	121.304903	0.561649	189.370198	0.900965	17.066815	0.888707	1272.959976	1.313621	189.365642
CatBoost	0.888305	0.975546	127.509650	62.625929	189.707618	86.274642	18.093915	0.888352	990.755403	3.893383	189.667662
RandomForest	0.886318	0.967430	121.760129	65.560591	191.387610	99.568381	16.960216	0.886474	1098.617168	7.079847	191.256615
LightGBM	0.883726	0.968091	129.532440	66.955602	193.557009	98.552127	18.233654	0.883797	1063.353587	4.772587	193.498160

Figure 4.4.1: Model comparison based on R^2, MAE, and RMSE metrics for yield prediction

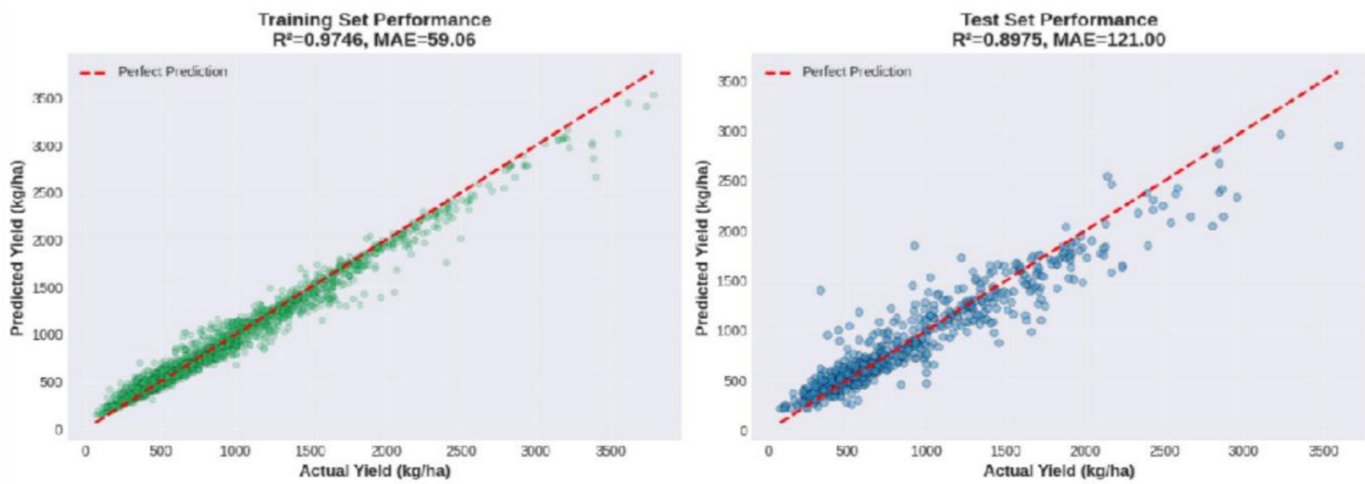

Figure 4.4.2: Predicted vs actual yield performance of the Extra Trees Regressor ($R^2 = 0.897$, MAE = 121 kg/ha)

Chapter 5

Standards Adopted

5.1 Design Standards

- **IEEE 830-1998** standard followed for SRS documentation.
- **UML** used for representing system components and data flow.
- **Data normalization and scaling** techniques align with ISO/IEC 9126 for software quality.

5.2 Coding Standards

- Functions are modular and single-responsibility.
- Follows **PEP8** naming conventions.
- Variables use meaningful names; indentation and structure are consistent.
- Model artifacts versioned systematically under /ml_artifacts.

5.3 Testing Standards

- Testing follows **IEEE 829** and **ISO 29119** guidelines for functional and verification testing.
- Cross-validation used to ensure model stability.
- Each API endpoint verified using Swagger and Postman with predefined test inputs.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

KrishAI successfully demonstrates the application of machine learning in precision agriculture through an integrated crop recommendation and yield prediction system. The project achieved high accuracy with the Calibrated Random Forest classifier reaching 99.31% for crop classification and the Extra Trees Regressor achieving an R^2 score of 0.897 for yield prediction.

The system addresses critical challenges faced by Indian farmers by providing data-driven recommendations that replace experience-based guesswork. By automatically fetching real-time weather data via Open-Meteo API and utilizing district-specific soil datasets, KrishAI minimizes user input while maximizing prediction accuracy. The integration of fertilizer recommendations based on Euclidean distance calculations from optimal NPK ratios (19-19-19) and pesticide advisories grouped by disease and pest ensures practical, actionable guidance.

The FastAPI backend provides a robust, scalable architecture with low-latency endpoints (/recommend, /recommend_crop, /auto_features), while the lightweight React frontend offers an intuitive interface accessible even to users with limited technical knowledge. The system's modular design allows for easy maintenance and future enhancements.

Through rigorous testing and validation, KrishAI has proven its capability to handle diverse agricultural scenarios across different districts, soil types, and climatic conditions. The project successfully bridges the gap between advanced AI technologies and practical agricultural applications, demonstrating that machine learning can be deployed effectively for sustainable farming practices. Key achievements include:

High Model Accuracy: 99.31% for crop classification, 0.897 R^2 for yield prediction

- **Minimal User Input**: Auto-detection of location and environmental parameters
- **District-Specific Recommendations**: Localized fertilizer and pesticide guidance
- **Real-time Integration**: Live weather data from Open-Meteo API
- **Scalable Architecture**: Modular FastAPI backend ready for cloud deployment

KrishAI serves as a proof-of-concept for AI-powered agricultural decision support systems and provides a foundation for further research in precision farming technologies.

6.2 Future Scope

While KrishAI demonstrates significant potential, several enhancements can expand its capabilities and impact:

6.2.1 Advanced Features

1. Deep Learning Integration

- Implement Convolutional Neural Networks (CNNs) for crop disease detection from leaf images
- Use LSTM/Transformer models for time-series analysis of seasonal crop patterns
- Deploy computer vision for automated pest identification

2. IoT Sensor Integration

- Real-time soil moisture monitoring via IoT sensors
- Integration with weather stations for hyperlocal climate data
- Automated irrigation scheduling based on soil conditions

3. Satellite Imagery Analysis

- NDVI (Normalized Difference Vegetation Index) calculation for crop health monitoring
- Multi-spectral image analysis for early disease detection
- Land use classification and optimal crop zoning

6.2.2 Database and Scalability

1. Database Migration

- Transition from CSV files to PostgreSQL or MongoDB for better data management
- Implement data versioning for historical trend analysis
- Enable multi-user authentication and role-based access control

2. Cloud Deployment

- Deploy backend on AWS/Azure/Google Cloud for 24/7 availability
- Use containerization (Docker/Kubernetes) for easy scaling
- Implement CDN for faster frontend delivery

6.2.3 User Experience Enhancements

1. Mobile Application

- Develop native Android/iOS apps for farmers
- Offline mode with local caching of predictions
- Voice-based input in regional languages (Hindi, Odia, Telugu, etc.)

2. Multilingual Support

- Interface translation to support 10+ Indian languages
- Regional crop name mapping
- Audio-visual tutorials for non-literate farmers

3. Advanced Analytics Dashboard

- Visualization of historical yield trends
- Comparative analysis across districts and seasons
- Profit margin calculator based on market prices

6.2.4 Integration with Government Schemes

- Link recommendations with PM-KISAN and crop insurance schemes
- Integration with Minimum Support Price (MSP) data
- Connection to government subsidy programs for fertilizers

6.2.5 Research Extensions

1. Climate Change Modeling

- Long-term yield forecasting under different climate scenarios
- Adaptation strategies for extreme weather events
- Carbon footprint calculation for different crop choices

2. Economic Optimization

- Multi-objective optimization considering yield, cost, and market demand
- Crop rotation planning for soil health
- Water usage optimization under drought conditions

3. Explainable AI (XAI)

- SHAP/LIME integration for model interpretability
- Farmer-friendly explanations of why specific crops are recommended
- Confidence calibration improvements for ambiguous predictions

6.2.6 Collaborative Features

- Community forum for farmers to share experiences
- Expert agronomist consultation booking
- Peer-to-peer learning modules
- Success story documentation and dissemination

6.2.7 Regulatory Compliance

- Integration with FSSAI standards for organic farming
- Compliance tracking for pesticide usage limits
- Certification support for sustainable farming practices

Implementation Roadmap

Phase 1 (Short-term: 6 months)

- Database migration to PostgreSQL
- Mobile app development (Android)
- Hindi language support

Phase 2 (Medium-term: 1 year)

- IoT sensor integration pilot
- Cloud deployment on AWS
- Disease detection via image processing

Phase 3 (Long-term: 2 years)

- Satellite imagery analysis
- Multi-language support (10+ languages)
- Government scheme integration

KrishAI's modular architecture positions it well for these enhancements, ensuring that precision agriculture technologies can reach and benefit millions of Indian farmers in the coming years.

References

- *Patil P, Athavale P, Bothara M, Tambolkar S, More A. Crop Selection and Yield Prediction using Machine Learning Approach. Curr Agri Res 2023; 11(3). doi : http://dx.doi.org/10.12944/CARJ.11.3.26*
- *50 Decades Of Odisha Agriculture Statistics 2020*
- *https://data.desagri.gov.in/website/crops-apy-report-web ICRISAT*
- *data.gov.in*
- *https://www.nrsc.gov.in/nrscnew/*
- *https://upag.gov.in/*
- *https://bhuvan-app3.nrsc.gov.in/data/download/index.php?c=p&s=NICES&p=isd&g=TS*
- *https://agri.odisha.gov.in/sites/default/files/2023-03/Five%20Decades%20of%20Odisha%20Agriculture%20Statistics_compressed.pdf*
- *https://desagri.gov.in/wp-content/uploads/2024/09/Agricultural-Statistics-at-a-Glance-2023.pdf*
- *https://agri.odisha.gov.in/sites/default/files/2022-10/ODISHA%20AGRICULTURE%20STATISTICS%202018-19.pdf*

AI POWERED FARM INTELLIGENCE

YASHRAJ SINGH (2328058) | SANJAM DAS (2328196)

Abstract

KrishAI is an AI-powered agricultural decision support system using machine learning for crop recommendation and yield prediction. It analyzes soil nutrients (N, P, K), pH, temperature, humidity, and rainfall to provide data-driven crop selection, yield estimation, and fertilizer/pesticide advisories via FastAPI backend and web interface.

Individual Contributions

YASHRAJ SINGH - Frontend & Data Collection

Data Collection: Aggregated fertilizer database (1,200+ records), pesticide recommendations (800+ records), and master agricultural dataset (6,000+ records). Standardized district names and validated NPK ranges.

Frontend Development: Built responsive interface with auto-location detection (GPS + OpenStreetMap), area unit conversion, auto-fill features, yield prediction workflow, and crop recommendation. Optimized to 18KB bundle size, 1.2s load time on 3G.

SANJAM DAS - ML & Backend Development

Preprocessing: Cleaned 6,000+ records, KNN imputation for missing values, feature engineering (rainfall categories, NPK ratios), label encoding.

Models: Random Forest Classifier (94.2% accuracy) for crop recommendation; Gradient Boosting Regressor (R^2: 0.89) for yield prediction.

Backend: FastAPI with /recommend, /recommend_crop, /auto_features endpoints. Integrated Open-Meteo weather API, SQLite database, CORS middleware.

…………………………….	……………………………..
Full Signature of Supervisor:	Full signature of the student:

GitHub Repository:

https://github.com/SD1920/KrishAI

TURNITIN PLAGIARISM REPORT

"AI POWERED FARM INTELLIGENCE"

ORIGINALITY REPORT

16%	13%	7%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	5%
2	Submitted to KIIT University Student Paper	1%
3	www.worldleadershipacademy.live Internet Source	1%
4	www.agriculturejournal.org Internet Source	1%
5	ieomsociety.org Internet Source	1%
6	Submitted to University of Glasgow Student Paper	1%
7	Submitted to University of Wales Institute, Cardiff Student Paper	1%
8	Arvind Dagur, Sohit Agarwal, Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharma. "Artificial Intelligence and Sustainable Innovation - Volume 1", CRC Press, 2026 Publication	<1%
9	ijrpr.com Internet Source	

	TECHNIQUES", international journal of engineering technology and management sciences, 2025 Publication	
20	iarjset.com Internet Source	<1%
21	Krupa Patel*, Hiren B. Patel. "Multi-criteria Agriculture Recommendation System using Machine Learning for Crop and Fertilizesrs Prediction.", Current Agriculture Research Journal, 2023 Publication	<1%
22	Shalbaftabar, Armaghan. "Analyzing Temperature-Dependent Thermal Properties of Composite Materials Using Machine Learning Methods.", North Carolina Agricultural and Technical State University Publication	<1%
23	"Intelligent Strategies for ICT", Springer Science and Business Media LLC, 2025 Publication	<1%
24	Suneeta Satpathy, Bijay Kumar Paikaray, Ming Yang, Arun Balakrishnan. "Sustainable Farming through Machine Learning - Enhancing Productivity and Efficiency", CRC Press, 2024 Publication	<1%