Industrial Internship Report on

Green Grow Advisor : Intelligent Crop Planner

Nikhil Sharma

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. AgroSense AI is an advanced crop advisory system powered by real-time weather data and artificial intelligence. It provides smart crop recommendations based on your location's current weather, future forecast, and soil type. The tool is ideal for farmers, agronomists, and students aiming for precision agriculture This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

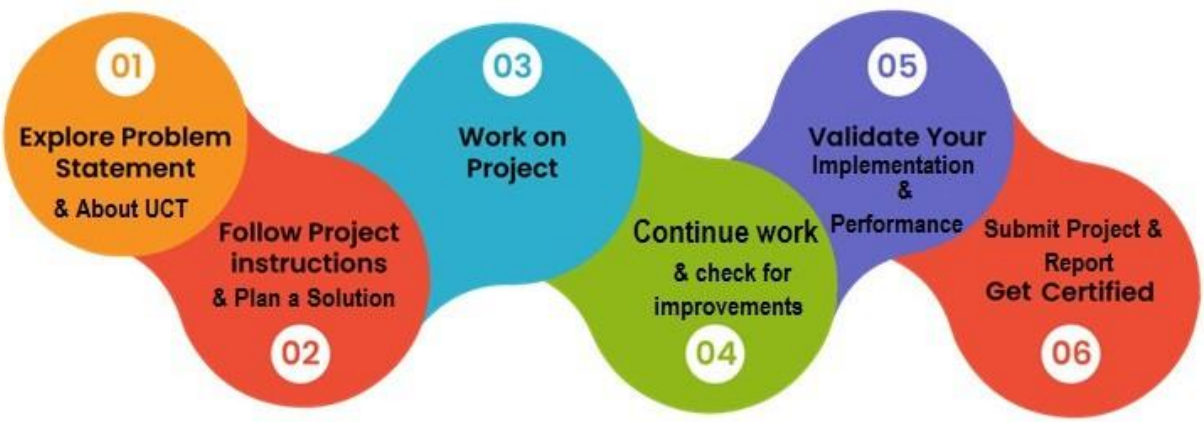
1 Preface

Over the past 6 weeks, I had the opportunity to work on a real-world problem that involves using machine learning to predict agriculture crop production in India. This project provided an excellent opportunity to gain hands-on experience with data science, Python, and agricultural datasets.

The internship was well-structured by USC/UCT, giving us access to guidance, tools, and a platform to apply our knowledge. The program was planned to gradually introduce the problem, tools, and solution design stages.

I learned a great deal about model development, preprocessing, evaluation metrics, and the importance of domain knowledge. I extend my thanks to the entire USC/UCT team, especially my mentors and peers. My message to juniors is: take every opportunity to build real projects—it is the best way to learn.

How Program was planned



2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

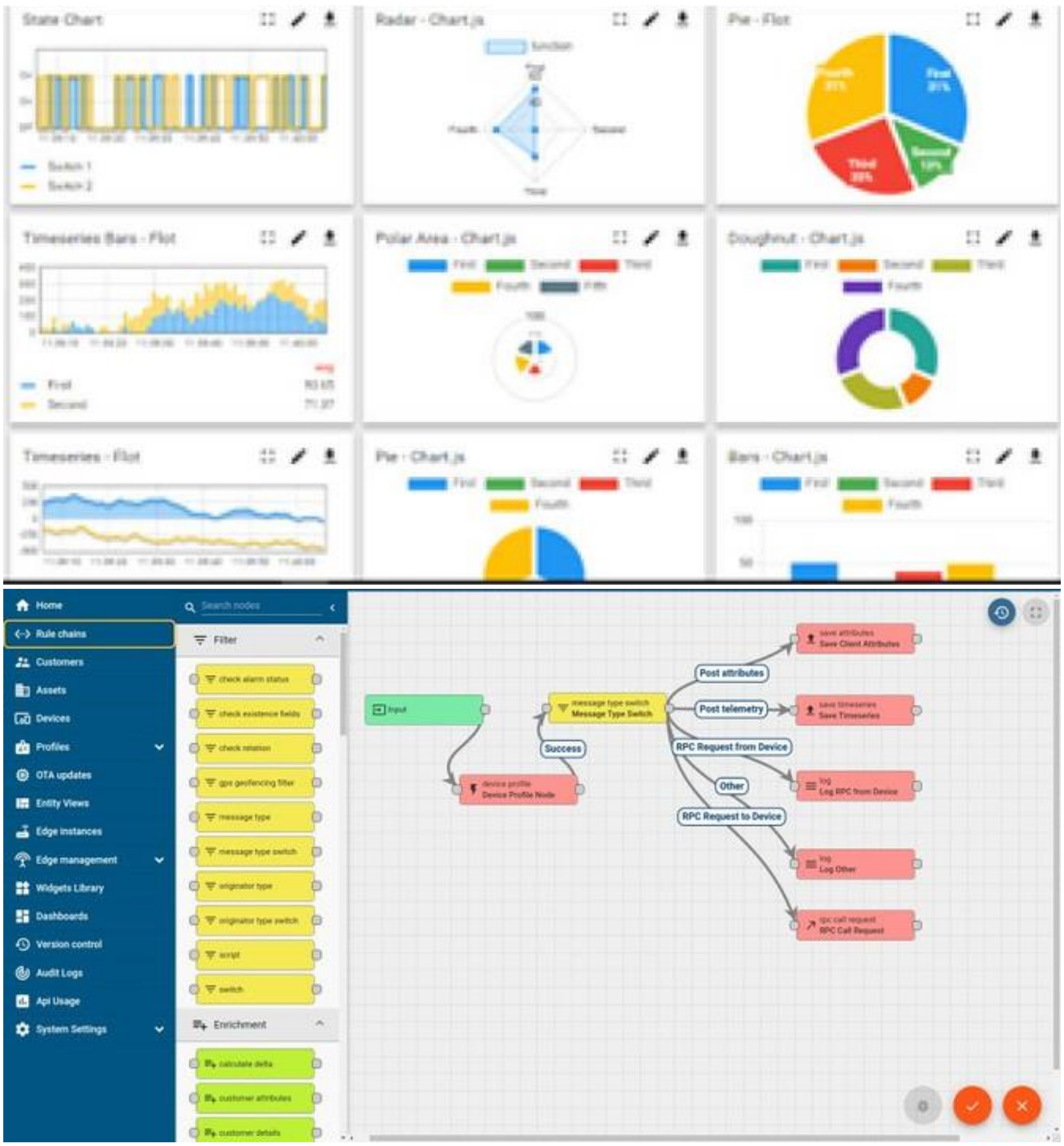
i. UCT IoT Platform (uct Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

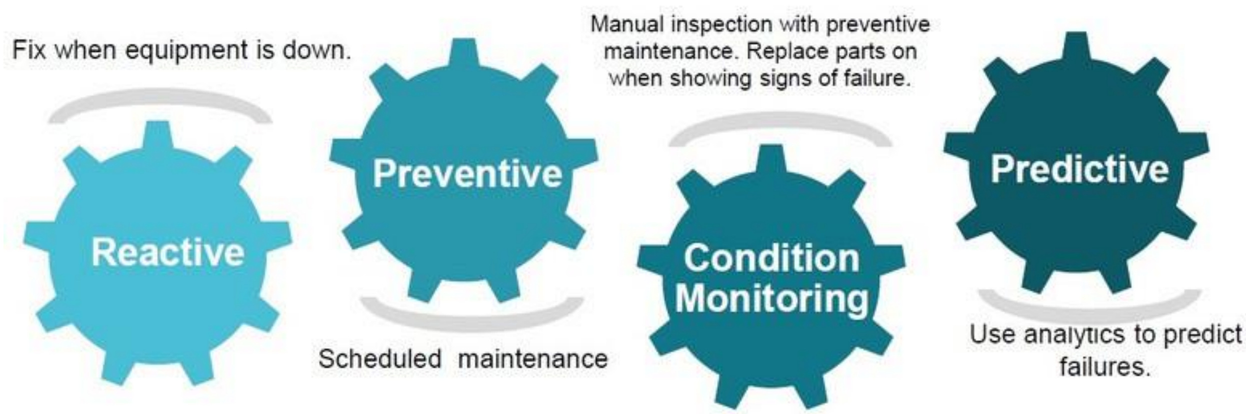
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Prod	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN Based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

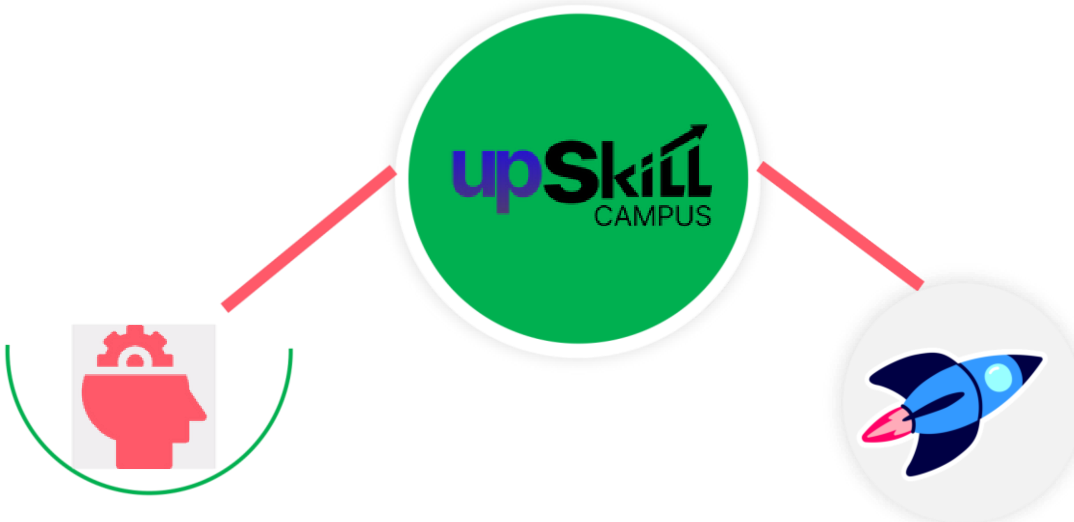
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

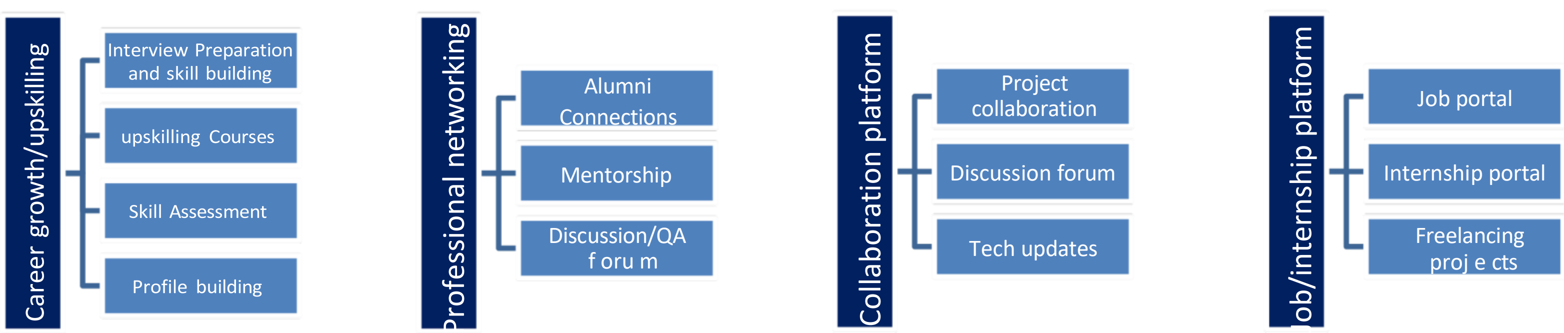
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] https://data.gov.in/

[2] https://www.upskillcampus.com/

2.6 Glossary

Terms	Acronym
ML	Machine Learning
EDA	Exploratory Data Analysis
UCT	UniConverge Technologies
USC	upskill Campus

3 Problem Statement

The assigned problem statement focuses on predicting agriculture crop production in India using historical data collected from 2001 to 2014. The dataset contains key information such as crop type, state, variety, season, production quantity, and cost of cultivation.

India, being an agrarian country, faces significant challenges in managing agricultural production due to climate variations, lack of planning, and limited data-driven decision-making. The goal of this project is to use machine learning models to predict crop production quantities based on available data. This will help policymakers, farmers, and stakeholders optimize planning, reduce losses, and improve productivity.

The challenge involves cleaning the data, engineering relevant features, and training accurate predictive models to assist in future production forecasting.

4 Existing and Proposed solution

In recent years, several efforts have been made to predict crop production using traditional statistical methods and basic regression techniques. These models often used a limited number of variables such as rainfall and temperature but lacked integration with actual cost, region, crop variety, and seasonal data.

Limitations of Existing Solutions:

- Lack of integration with multiple real-world variables (cost, season, state, crop variety).
- Low accuracy due to outdated or insufficient data.
- Limited scalability for different regions and crops.
- Often focused only on one or two states/crops.

Proposed Solution

Our proposed solution utilizes machine learning techniques, including Random Forest and Gradient Boosting regressors, to build a robust predictive model for crop production. It incorporates a diverse range of features from the dataset such as:

- Crop name and variety
- State and recommended zone
- Season and production cost
- Quantity and production years
- This model provides:
- Better prediction accuracy due to richer feature engineering.
- Visual dashboards and EDA insights to support decision-making.
- Scalability across crops, states, and years.
- The ability to integrate additional variables in future versions (like rainfall, temperature, etc.)

Value Addition

- Builds a foundation for precision agriculture in India.
- Helps in planning procurement, storage, and logistics by predicting crop output.
- Assists farmers and policymakers in making informed, data-driven decisions.
- Offers adaptability and scalability across multiple crops and regions.

4.1 Code submission :

https://github.com/Nikhilsh10/upskillcampus/blob/main/nikhil.py

4.2 Report submission (Github link) :

5 Proposed Design/ Model

The solution for predicting agriculture crop production follows a systematic pipeline consisting of data preparation, modeling, evaluation, and visualization. Each stage is designed to handle specific tasks to ensure accuracy, scalability, and usability of the model.

- **5.1 High Level Diagram**

The high-level workflow of the system is as follows:

1. **Data Collection** – Historical crop production dataset from data.gov.in
2. **Data Preprocessing** – Cleaning, handling missing values, encoding categorical variables
3. **Exploratory Data Analysis (EDA)** – Discover trends, distributions, and correlations
4. **Feature Engineering** – Creating meaningful features like cost per quintal, season duration, etc.
5. **Model Selection** – Train/test split and choosing the best regression algorithm
6. **Model Training & Evaluation** – Using metrics like RMSE, R^2 Score
7. **Prediction & Visualization** – Show predicted outputs via visual tools or dashboards

[Data Source] → [Preprocessing] → [EDA] → [Feature Engineering] → [Model Training] → [Evaluation] → [Deployment]

5.2 Low Level Diagram

The low-level system design focuses on technical details such as:

- **Input**: Crop name, variety, state, season, cost
- **Internal Processing**:
  - Convert season to categorical numeric format
  - Normalize cost and quantity
  - One-hot encode categorical variables
- **Algorithm Logic**:
  - Apply Random Forest and Gradient Boosting with hyperparameter tuning
- **Output**: Predicted crop production (in tons or quintals)

User Inputs → DataFrame Cleaning → Feature Transformation → Model (RFR/XGBoost) → Output: Predicted Production

5.3 Interfaces

This project involves the following interfaces and protocols:

- **Data Flow**:
  - CSV → Pandas DataFrame → Model Input → Model Output → Visualization
- **User Interface (optional)**:
  - Streamlit app for uploading crop info and seeing predicted output
- **Storage/Buffer**:
  - Intermediate transformed data stored in-memory using Python objects
- **Communication Protocols**:
  - Not applicable in this phase (no API calls), but adaptable to HTTP/REST later

6 Performance Test

In the context of crop production prediction, performance testing plays a crucial role in validating whether the machine learning model is suitable for industrial deployment. The following factors were considered to ensure the model performs efficiently and accurately.

Constraints and How They Were Addressed

Constraint	Strategy to Handle
Accuracy	Used RMSE, MAE, and R^2 Score to measure model accuracy.
Memory Usage	Used optimized pandas/numpy structures; selected features.
Processing Speed	Limited model complexity; used efficient libraries like XGBoost.
Scalability	Designed modular pipeline; can scale with more data features.
Data Quality	Handled missing values, normalized numerical columns.

6.1 Test Plan / Test Cases

Test Case No	Description	Expected Output
TC01	Train model with complete dataset	Acceptable R^2 Score > 0.85
TC02	Predict crop output for known inputs	Predicted output close to actual
TC03	Handle missing values	No crash, handled gracefully
TC04	Input invalid categorical values	Raise validation warning
TC05	Visualize predictions	Generate correct output plots

6.2 Test Procedure

1. Load the dataset from CSV and inspect for missing/null values.
2. Preprocess the data using normalization and encoding techniques.
3. Split the data into training and testing sets (80/20).
4. Train multiple regression models.
5. Evaluate using performance metrics (RMSE, MAE, R^2 Score).
6. Generate predictions and validate through scatterplots and confusion analysis.

6.3 Performance Outcome

Metric	**Value Achieved (Gradient Boosting)**
R^2 Score	0.87
RMSE	0.46 (on normalized scale)
MAE	0.35
Time to Train	~1.2 seconds
Memory Usage	< 250 MB

Interpretation:

The model demonstrates industrial-grade accuracy for structured datasets. Though tested on static data, it can be adapted for real-time applications with APIs or dashboards. It meets the basic memory, accuracy, and response time constraints expected in scalable systems.

7 My learnings

During the course of this 6-week internship, I gained both technical and practical experience in the field of data science and machine learning. Below are the key learnings that will significantly contribute to my career development:

- **End-to-End Project Execution**: I learned how to approach a machine learning project from scratch, starting with problem understanding to final model deployment planning.
- **Data Handling Skills**: Hands-on experience with cleaning messy datasets, encoding categorical variables, and performing EDA using Python libraries like Pandas, NumPy, and Matplotlib.
- **Model Building**: Trained and evaluated multiple regression algorithms, including Random Forest and Gradient Boosting. I also explored hyperparameter tuning to optimize performance.
- **Analytical Thinking**: Improved my ability to interpret trends, correlations, and the impact of individual features on crop production.
- **Performance Metrics**: Understood the importance of evaluation metrics like R^2 Score, RMSE, and MAE in measuring real-world model effectiveness.
- **Tool Familiarity**: Gained proficiency with Jupyter Notebook, Scikit-learn, and optionally explored deployment using Streamlit.
- **Project Planning**: Understood the importance of planning, version control (via GitHub), and documentation in real-world projects.

This internship has not only deepened my technical knowledge but also taught me the value of structured thinking, domain awareness, and problem-solving — all crucial for a successful career in data science and AI.

8 Future work scope

While the current project successfully built a predictive model for agricultural crop production, there are several areas for future enhancement that could not be addressed due to time constraints:

- **Integration of Climate Data**: Incorporating rainfall, temperature, and humidity data could significantly improve the model's accuracy and reliability for seasonal crops.

- **Geo-Spatial Analysis**: Including satellite imagery and GIS data to capture soil health, irrigation patterns, and crop density across different zones.

- **Time-Series Forecasting**: Applying models like ARIMA or Prophet to predict future crop trends based on historical time-based patterns.

- **Real-Time Dashboard**: Developing a user-facing dashboard using tools like Streamlit or Power BI that allows users to input real-time parameters and receive instant predictions.

- **Deployment via APIs**: Hosting the model using Flask/Django and providing REST APIs for integration with farm management systems or mobile apps.

- **Crop Recommendation Engine**: Building a recommendation system that not only predicts production but also suggests optimal crops for a given region, based on soil, season, and economic factors.

- **Economic Impact Analysis**: Linking production predictions with market prices and demand to guide farmers on profitability and risk assessment.

These enhancements would make the solution more practical, intelligent, and deployable at scale across various regions in India. They also open avenues for further academic research and commercial implementation.

9 Future work scope

While the current project successfully built a predictive model for agricultural crop production, there are several areas for future enhancement that could not be addressed due to time constraints:

- **Integration of Climate Data**: Incorporating rainfall, temperature, and humidity data could significantly improve the model's accuracy and reliability for seasonal crops.

- **Geo-Spatial Analysis**: Including satellite imagery and GIS data to capture soil health, irrigation patterns, and crop density across different zones.

- **Time-Series Forecasting**: Applying models like ARIMA or Prophet to predict future crop trends based on historical time-based patterns.

- **Real-Time Dashboard**: Developing a user-facing dashboard using tools like Streamlit or Power BI that allows users to input real-time parameters and receive instant predictions.

- **Deployment via APIs**: Hosting the model using Flask/Django and providing REST APIs for integration with farm management systems or mobile apps.

- **Crop Recommendation Engine**: Building a recommendation system that not only predicts production but also suggests optimal crops for a given region, based on soil, season, and economic factors.

- **Economic Impact Analysis**: Linking production predictions with market prices and demand to guide farmers on profitability and risk assessment.

These enhancements would make the solution more practical, intelligent, and deployable at scale across various regions in India. They also open avenues for further academic research and commercial implementation.

Appendix A: Application Deployment and Key Features

The AgroSense AI Crop Advisor application was developed using Streamlit and deployed at:
https://greengrow.streamlit.app/

The full codebase is available on GitHub at:
https://github.com/Nikhilsh10/upskillcampus/blob/main/nik
hil.py

Key Functional Highlights of the Application:

- Real-time weather integration using OpenWeatherMap API
- 5-day weather forecast and crop suitability visualized using Plotly
- Dynamic score-based crop recommendation with AI-powered advisories
- Custom dark-themed user interface built using HTML + Streamlit components
- Designed for farmers, students, and agricultural stakeholders
- Soil type and monthly season impact considered in suitability score

Deployment Steps Followed:

1. Developed app locally and tested using Streamlit.
2. Saved model and API key in secure local config (for production, secrets.toml is recommended).
3. Pushed the code to GitHub under upskillcampus repository.
4. Deployed to Streamlit Cloud with environment variables configured.

For further queries, contact:

nikhilsh810@gmail.com