Project Report: Precision Agri-Advisor

AI-Driven Crop Recommendation System for Sustainable Agriculture

Author: [Sudipta Biswas]
Date: [05-12-2025]

1. Executive Summary

The **Precision Agri-Advisor** is a Machine Learning-based decision support system designed to assist agronomists and farmers in selecting the optimal crop for specific soil and climatic conditions. By analyzing key environmental parameters—Nitrogen (N), Phosphorus (P), Potassium (K), pH, temperature, humidity, and rainfall—the system recommends the most suitable crop with **99% accuracy**.

The model was deployed as a live web application using **Streamlit**, demonstrating the ability to translate complex data models into accessible, user-friendly business tools. This project aligns with the principles of **Precision Agriculture**, aiming to maximize yield while minimizing resource wastage.

2. Problem Statement

The Challenge:
In traditional farming, improper crop selection based on intuition rather than data often leads to:

- Sub-optimal yields.
- Excessive use of fertilizers (damaging soil health).
- Economic loss for farmers.

The Solution:
A data-driven recommendation engine that leverages historical agricultural data to predict the best crop fit, ensuring biological feasibility and economic sustainability.

3. Data Overview

Source: Crop Recommendation Dataset (Kaggle).
Volume: 2,200 data points covering 22 unique crops (Rice, Maize, Coffee, Jute, Cotton, etc.).
Feature Set:

1. **Nitrogen (N):** Ratio of Nitrogen content in soil.
2. **Phosphorus (P):** Ratio of Phosphorus content in soil.
3. **Potassium (K):** Ratio of Potassium content in soil.
4. **Temperature:** Average temperature in °C.
5. **Humidity:** Relative humidity in %.
6. **pH:** pH value of the soil (Acidic vs. Alkaline).

7. **Rainfall:** Annual rainfall in mm.

Quality Control:

- Checked for null values (None found).
- Checked for duplicates (None found).
- Verified data types (Numerical inputs, Categorical target).

4. Methodology & Feature Engineering

4.1. Exploratory Data Analysis (EDA)

Statistical analysis revealed that certain features, particularly **Rainfall** and **Potassium**, exhibited high skewness. Understanding these distributions was critical for selecting the right preprocessing techniques.

4.2. Feature Engineering (Domain Logic)

To improve model interpretability and performance, I engineered new features based on agronomic principles:

- **Total Nutrients:** Sum of N, P, and K.
- **Nutrient Ratios:** Calculated N_ratio, P_ratio, and K_ratio.
  - *Why:* Crops rely on the *balance* of nutrients, not just absolute values. This helps the model distinguish between crops with similar total nutrient needs but different ratio requirements.

4.3. Data Preprocessing Pipeline

I implemented a robust pipeline using Scikit-Learn:

- **Label Encoding:** Converted target crop names (e.g., "Rice") into numerical labels.
- **Power Transformer (Yeo-Johnson):** Applied to numerical features to reduce skewness and make the data more Gaussian-like.
- **Standard Scaler:** Standardized features to ensure that variables with larger scales (like Rainfall) did not dominate variables with smaller scales (like pH).

5. Model Development & Selection

I adopted a **Champion vs. Challenger** approach to select the best algorithm.

5.1. Models Tested

1. **Decision Tree:** (Baseline) - Interpretable but prone to overfitting.
2. **Gradient Boosting:** (Challenger) - High performance but computationally expensive.
3. **Random Forest Classifier:** (Champion) - Ensemble method that reduces variance and overfitting.

5.2. Evaluation Metrics

The models were evaluated on Accuracy, Precision, Recall, and F1-Score.

Model	Accuracy	Result
Decision Tree	~96%	Good Baseline
Random Forest	**99.3%**	**Selected**
Gradient Boosting	98.9%	Strong Contender

Why Random Forest?
It effectively handled the non-linear relationships between weather patterns and crop growth. As an ensemble of decision trees, it provided the highest stability and generalization capability.

6. Deployment Architecture

The technical implementation focused on reproducibility and accessibility.

- **Language:** Python 3.10+
- **Libraries:** Pandas, Scikit-learn, Seaborn, Matplotlib.
- **Frontend framework:** Streamlit (for rapid UI development).
- **Serialization:** joblib was used to save the trained model (.pkl), label encoder, and feature names.
- **Hosting:** Deployed on **Streamlit Community Cloud** linked directly to a **GitHub Repository**.

Application Features:

1. **Interactive Sliders:** Allows agronomists to input soil test report data easily.
2. **Real-Time Inference:** Generates predictions instantly using the cached model.
3. **Business Logic Layer:** Displays "Agronomist Notes" alongside predictions (e.g., warning about high moisture needs for Rice), adding qualitative value to the quantitative prediction.

7. Results & Business Impact

- **Accuracy:** The system successfully identifies the correct crop 99 times out of 100 on test data.
- **Usability:** The web interface requires zero technical knowledge to operate, making it accessible to field officers.
- **Sustainability:** By recommending the "right crop for the right soil," the system supports UPL's mission of sustainable agriculture by potentially reducing the need for corrective

chemical interventions later in the season.

8. Conclusion

This project demonstrates the successful application of Data Science to a real-world agricultural problem. It covers the full lifecycle of a data project: from cleaning raw data and applying statistical feature engineering to training high-performance models and deploying a user-facing product.

Repository: [GitHub]
Live Demo: [Streamlit App]