## Verdantia : A Smart Gardening Assistant for Crop Risk Prediction and Rule-Based Advisory for Beginner Vegetable Growers


**Shaine Carla P. Macatangay, Ella Edz A. Taquilid**
College of Computer Studies
Laguna State Polytechnic University


**A Hybrid Machine Learning and Knowledge-Based Reasoning Project**
CSST101 – Machine Learning CSST102 – Knowledge Representation and Reasoning
**Instructor:** Mr. Mark P. Bernardino

**Date Submitted:** December 31, 2025


**Email:** shainecarla.macatangay@lspu.edu.ph



### Abstract

The Smart Gardening Assistant is a web-based decision support system designed to help beginner vegetable growers manage plant health through data-driven insights. The system focuses on predicting water-related plant conditions, specifically healthy, overwatered, and underwatered states, using machine learning techniques. Environmental and crop-related inputs such as crop type, temperature, humidity, rainfall, soil moisture, soil type, and sunlight exposure are analyzed by a classification model to determine the plant's condition.

To enhance interpretability and usability, the system integrates Knowledge Representation and Reasoning (KRR) through a rule-based reasoning approach. The rule-based module interprets the machine learning output and generates clear, actionable gardening recommendations. By combining predictive modeling with logical reasoning, the Smart Gardening Assistant provides beginner-friendly guidance that supports sustainable home gardening practices. The project contributes to food security and environmental responsibility, aligning with Sustainable Development Goals 2 (Zero Hunger), 11 (Sustainable Cities and Communities), and 13 (Climate Action).



Keywords: Smart Gardening, Machine Learning, Decision Support System, Knowledge Representation and Reasoning (KRR), Sustainable Agriculture

## I. PROJECT OVERVIEW

Beginner vegetable growers often struggle with proper watering due to limited experience in interpreting environmental conditions such as soil type, temperature, humidity, and rainfall. As a result, many rely on fixed watering schedules, which frequently lead to overwatering or underwatering, negatively affecting plant health, yield, and water efficiency.

To address this issue, the Smart Gardening Assistant is a web-based decision support system that integrates Machine Learning (ML) and Knowledge Representation and Reasoning (KRR). The ML component analyzes environmental and soil-related data to classify a plant’s water condition as Healthy, Overwatered, or Underwatered. The KRR component then applies rule-based logic to transform these predictions into clear and actionable watering recommendations tailored to the user’s context.

The system aims to help beginner gardeners make informed watering decisions, reduce plant stress and water waste, and promote sustainable gardening practices that support food security and environmental responsibility.

## II. OBJECTIVES

### General Objective

The general objective of the project is to develop a smart, web-based gardening assistant that predicts water-related plant conditions and generates rule-based recommendations to support beginner vegetable growers in making informed, sustainable watering decisions.

This overall goal combines predictive analytics with explainable reasoning so that users can both receive accurate assessments of their plants’ water status and understand the practical steps needed to correct or maintain those conditions. The system seeks to bridge the gap between raw environmental data and user-friendly guidance, enabling beginners to care for their crops more effectively while promoting responsible water and soil management.

**Specific Objectives**

**Objectives related to Machine Learning functionality**

- To train and evaluate a classification model (for example, a Random Forest classifier) capable of distinguishing between healthy, overwatered, and underwatered plant states with satisfactory accuracy and reliability for practical use.
- To generate interpretable model outputs, including class probabilities or confidence measures, that can be used by the reasoning module to determine when predictions are strong, uncertain, or require rule-based adjustment.

**Objectives related to knowledge-based reasoning**

- To design a set of rule-based logic statements that encode expert-inspired knowledge about the relationships between plant condition, soil type, rainfall, soil moisture, temperature, and other relevant factors.
- To integrate these rules into a reasoning engine that can interpret the Machine Learning predictions, detect critical edge cases (such as potential waterlogging in heavy soils or severe drought in sandy soils), and refine or override the predicted class when appropriate.

**Objectives related to system outcomes**

- To develop a user-friendly web interface that allows beginners to input crop and environmental data easily, view the predicted plant condition, and read the corresponding recommendations in a clear, non-technical format.
- To support sustainable home gardening by embedding recommendations that encourage efficient water use, improved soil health, and climate-conscious practices, thereby contributing to food security and environmental responsibility.

## III. SYSTEM ARCHITECTURE

**System Flow:**

**User → Web Interface → Mode Selection → Input Data →**
**Machine Learning / Rule-Based Processing →**
**Recommendations & Diagnosis → Output Display**

1. **User input**

   The system begins with a user input layer where gardeners provide soil and environmental information through a web-based form. Inputs include soil nutrients (N, P, K), temperature, humidity, soil pH, and rainfall, along with contextual data such as soil type and climate, and optional information about the current plant or user profile. These values collectively describe the conditions of the garden plot and form the feature vector passed to the Machine Learning model and the knowledge-based reasoning rules.

2. **Machine Learning model processing**

   A pre-trained scikit-learn multi-class classifier (loaded from a model file) processes the inputs to recommend suitable crops. If probability outputs are available, the system ranks the top five crops; otherwise, it uses a single predicted label and, when needed, a similarity-based fallback that compares the user's inputs to the dataset using normalized distances.

3. **Knowledge-based rules**

   Alongside crop recommendation, a deterministic rule-based module evaluates the same inputs to detect water-related risks (overwatering, underwatering, flood or waterlogging) and nutrient deficiency. Each IF–THEN rule produces a diagnosis label, a brief explanation, and practical advice on watering, drainage, mulching, or fertilization.

4. **Final decision and recommendations**

   The system then merges ML and rule-based outputs into two views: a recommendation view with a primary crop, alternatives, and growing tips, and a diagnosis view with the detected condition, its cause, and clear next steps. These results are displayed in the web interface so users can see what to plant, what risks exist, and what actions to take.

## IV. MACHINE LEARNING COMPONENT

### Algorithm Used

The Machine Learning component uses a scikit-learn multi-class classifier that exposes the predict_proba interface and classes_ attribute, enabling probability-based ranking of crops. Typical implementations for this dataset include tree-based methods such as Random Forest, Gradient Boosting, or similar classifiers that handle tabular NPK–weather features effectively.

### Dataset Size

The model is trained on a cleaned version of the Crop Recommendation dataset, which contains approximately 2,180 usable records after preprocessing, each representing a unique combination of soil nutrients and weather conditions mapped to a specific crop label.

### Model Accuracy or Evaluation Metric

While the repository does not store explicit metrics in code, standard practice for this dataset is to evaluate models using overall accuracy and macro-averaged F1-score, supported by a confusion matrix to understand per-class performance. Cross-validation, typically stratified by crop label, is recommended to assess robustness and reduce the risk of overfitting.

## V. MACHINE LEARNING PIPELINE

### Data Collection Method

Data is sourced from a local CSV file derived from the public Crop Recommendation dataset, with fields for N, P, K, temperature, humidity, soil pH, rainfall, and the target crop label. The CSV is loaded into a Python environment (for example, Google Colab) using pandas for further preprocessing and model training.

### Data Preprocessing Steps

Preprocessing includes reading and validating the tabular data, checking for missing or inconsistent values, and ensuring numeric types for all feature columns. For training, models may either use raw numeric features (for tree-based classifiers) or apply feature scaling or standardization (for linear models), while the deployed application additionally uses min–max scaling when computing fallback similarity rankings between user inputs and dataset rows.

### Model Training Approach

The training task is framed as supervised multi-class classification, with the feature vector [N,P,K,temperature,humidity,pH,rainfall]. It mapped to a categorical target representing crop name. The data is split into training and validation subsets, optionally using stratified K-fold cross-validation, and hyperparameters are tuned (for example, via grid search) to find a model that balances accuracy and generalization.

### Model Evaluation Method

Model performance is evaluated using accuracy, macro-F1, and per-class precision and recall, providing insight into how well the model predicts both majority and minority crop classes. A confusion matrix is generated to analyze misclassifications, and cross-validation metrics are used to verify stability across different data splits.

### Model Deployment Method

Once a satisfactory model is trained, it is serialized using a persistence library such as joblib and saved as a file (for example, crop_model.pkl). In deployment, a Flask-based web application loads the model at startup, receives user inputs from HTTP requests, passes them to the classifier's predict_proba or predict method, and returns the resulting crop recommendations to be rendered in the interface.

## VI. DATASET DESCRIPTION

### Dataset Type

The dataset is a structured, tabular multi-class classification dataset, where each row corresponds to a set of soil and weather measurements and the associated recommended crop. All predictor variables are continuous or numeric, and the target is a categorical label.

### Number of Records

The original Crop Recommendation dataset contains about 2,200 records; after cleaning and removal of invalid rows, the working dataset used for training comprises approximately 2,180 records.

### Target Variable

The target variable is the crop label, which indicates the most suitable crop for the given combination of N, P, K, temperature, humidity, soil pH, and rainfall.

## VII. KNOWLEDGE REPRESENTATION & REASONING

**Rule 1 (IF–THEN)**

IF climate is "Tropical Wet" AND soil type is "Clay" THEN condition = "Flood / Waterlogging Risk" AND advice = improve drainage and reduce watering to prevent root suffocation and prolonged saturation.

**Rule 2 (IF–THEN)**

IF rainfall is greater than a high threshold (for example, 200 mm in the recent period) THEN condition = "Overwatered" AND advice = reduce or temporarily stop watering and ensure excess water can drain away from the root zone.

**Rule 3 (IF–THEN)**

IF rainfall is below a low threshold (for example, less than 40 mm) THEN condition = "Underwatered" AND advice = increase watering, apply mulch to reduce evaporation, and monitor soil moisture more frequently.

**Rule 4 (IF–THEN)**

IF humidity is very high (for example, above 85%) AND temperature is high (for example, above 30 °C) THEN condition = "Pest Risk" AND advice = inspect leaves for pests, improve airflow, and consider gentle, organic pest control methods.

**Rule 5 (IF–THEN)**

IF any of N, P, or K is significantly below a defined deficiency threshold THEN condition = "Nutrient Deficiency" AND advice = apply a balanced organic fertilizer or soil amendment appropriate to the missing nutrient(s).

## VIII. HYBRID DECISION LOGIC

The system employs a hybrid decision-making framework that integrates Machine Learning (ML) with Knowledge Representation and Reasoning (KRR). The ML component processes soil and environmental features to generate probabilistic crop suitability predictions, which are used to identify a primary recommendation and rank alternative crops. In parallel, the KRR module applies domain-specific IF–THEN rules to evaluate contextual risks and growing conditions, producing diagnostic assessments and actionable advice. The final output combines ML-based recommendations with rule-based diagnoses into a unified decision package, ensuring both data-driven accuracy and knowledge-based interpretability. When model confidence is low or predictions are unavailable, the system gracefully degrades by applying similarity-based ranking while retaining rule-based reasoning, thereby maintaining reliable user guidance.

## IX. SYSTEM FEATURES

- Wellness risk prediction: The system provides qualitative diagnoses such as overwatered, underwatered, flood/waterlogging risk, pest risk, and nutrient deficiency based on rule evaluation of the input conditions.
- Rule-based recommendations: For each detected condition, users receive short, practical tips describing how to adjust watering, drainage, mulching, pest management, or fertilization.
- Web interface or API: A Flask-based web application exposes routes for home, crop recommendation, gardening advisor, and related functions, allowing users to interact through a browser and obtain real-time feedback.
- Google Colab deployment: Model training and evaluation can be performed in Google Colab using scikit-learn and pandas, with the trained model exported via joblib and then integrated into the local Flask app for deployment.

## X. TESTING AND EVALUATION

### Test case 1 – Pest risk

- Input summary: High temperature (around 32 °C), very high humidity (around 90%), moderate rainfall, loam soil, and tropical wet climate.
- Expected output: Diagnosis = “Pest Risk”; Advice = inspect leaves and stems for pests, improve airflow, and apply organic pest control if needed.

### Test case 2 – Underwatered

- Input summary: Low recent rainfall (for example, 30 mm), sandy soil, tropical dry climate, and typical warm temperatures.
- Expected output: Diagnosis = “Underwatered”; Advice = increase watering frequency, water more deeply, and add mulch to reduce moisture loss.

### Test case 3 – Flood / waterlogging risk

- Input summary: High rainfall (for example, 160 mm or more), clay soil, tropical wet climate, and moderate to high humidity.
- Expected output: Diagnosis = “Flood / Waterlogging Risk”; Advice = improve drainage (for example, raised beds or channels), avoid further watering until the soil dries, and monitor plants for root stress.

**Test case 4 – Healthy**

- Input summary: Mid-range values for rainfall, temperature, and humidity, with loam soil and no extremes in nutrient values.
- Expected output: Diagnosis = “Healthy”; Advice = maintain current watering schedule, continue monitoring conditions, and apply general good-practice tips such as mulching and occasional soil checks.

In all cases, the ML component additionally returns a crop recommendation and alternatives consistent with the input conditions, which are expected to align with known suitability patterns in the dataset.

## XI. CONCLUSION

### Summary of system results

The Smart Gardening Assistant successfully integrates a Machine Learning crop recommendation model with a rule-based Knowledge Representation and Reasoning module to support beginner gardeners. The system produces both crop suggestions and wellness diagnoses, accompanied by concrete, easy-to-follow advice that helps users respond to overwatering, underwatering, and nutrient issues.

### Key findings

The combination of probability-based crop ranking and deterministic rules provides robustness, ensuring that users receive meaningful guidance even when model confidence is low or the model is unavailable. Encoding expert-inspired rules makes the recommendations more transparent and safety-focused, particularly around water management, which are common sources of failure for beginners.

### Possible improvements

Future work could include explicitly logging and displaying training metrics alongside the model, adding feature-importance or SHAP-based explanations to show which inputs most influenced each recommendation, and expanding the rule base to capture more crops, climates, and management strategies. Additional improvements might integrate user feedback loops, regional calibration of thresholds, and richer sensor inputs, further enhancing accuracy, personalization, and educational value for home gardeners.

| **Member Name** | **Contribution** |
| --- | --- |
| Macatangay, Shaine Carla P. | Machine Learning model development, ML pipeline design and implementation, dataset collection, cleaning, and preprocessing, integration of ML outputs into the system, user interface (UI) design and implementation, project documentation. |
| Taquilid, Ella Edz A. | Knowledge Representation and Reasoning (KRR) rule design and implementation, user interface (UI) design, project documentation. |

## REFERENCES


Ingle, A. (2020, December 18). *Crop recommendation dataset*. Kaggle. https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

DebuggerCafe. (2024, September 14). *PlantVillage dataset: Disease recognition using PyTorch*. https://debuggercafe.com/plantvillage-dataset-disease-recognition-using-pytorch/

Food and Agriculture Organization of the United Nations. (n.d.). *FAO country profiles: Philippines*. https://www.fao.org/countryprofiles/index/en/?iso3=phl

Philippine Statistics Authority. (2012, July 30). *Philippine Food Security Information System (PhilFSIS)*. PSA OpenSTAT. https://openstat.psa.gov.ph/Featured/philfsis

United Nations Department of Economic and Social Affairs. (2025, October 24). *Sustainable Development Goal 2: Zero hunger*. https://sdgs.un.org/goals/goal2

United Nations. (2020, July 10). *Sustainable Development Goal 13: Climate action*. Wikipedia. https://en.wikipedia.org/wiki/Sustainable_Development_Goal_13

United Nations. (2024, December 31). *The 17 Sustainable Development Goals*. https://sdgs.un.org/goals