Crop Recommendation System Using Machine Learning

By

<22BCE3926> < HARSHITA LALWANI>

<22BCE2020> <ADITYA MANAN>

<22BCE0603> <VYASHNAVI KRISHNA>

Under the Guidance of

Dr. Swarnalatha. P

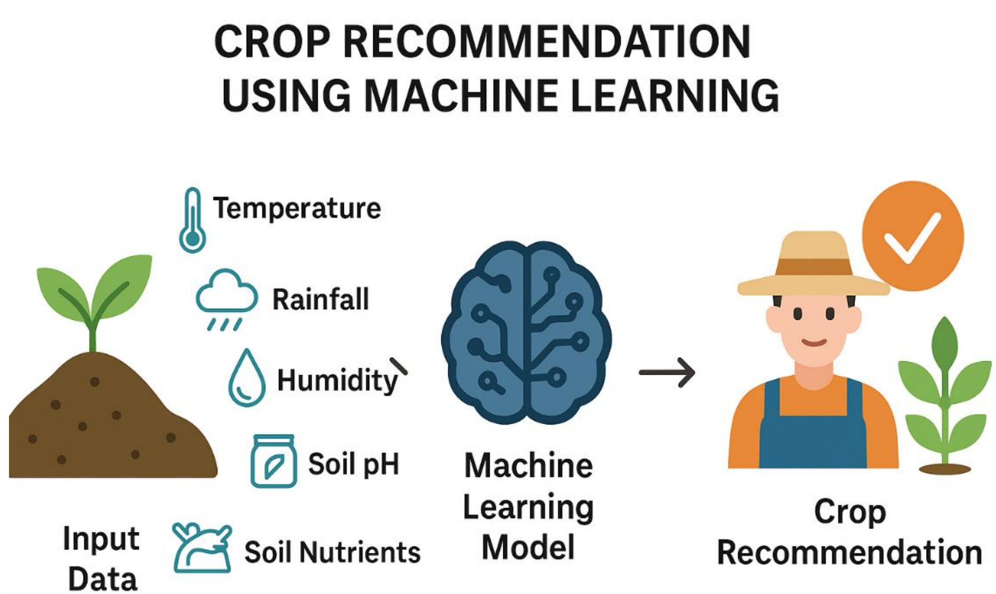
School of Computer Science and Engineering

BCSE316L-

Desing of Smart Cities

Fall Semester 2022

Review - I



Project Details

1. **Introduction & Problem Statement**
2. **Project Objective**
3. **Dataset Overview**
4. **Data Exploration & Analysis**
5. **Data Preprocessing**
6. **Machine Learning Models**
7. **Model Training Strategy**
8. **Model Performance Results**
9. **Selecting the Best Model**
10. **Methodology**
11. **Implementation**
12. **Result and Discussion**
13. **Conclusion & Impact**

- Course: BCSE316L – Design of Smart Cities (Winter Semester 2025)
- Project Title: Crop Recommendation Using Machine Learning
- Team Members: [Add Names and Registration Numbers]
- Under the Guidance of: Dr. Swarnalatha. P School of Computer Science and Engineering

INTRODUCTION AND PROBLEM STATEMENT

- Assistance
- Prediction
- Optimization
- Sustainability
- Efficiency
- Profitability
- Decision-Support
- Data-Driven
- Web-Based

- **The Challenge:** Farmers often struggle to select the most profitable and suitable crop for their specific land.
- **Current Problem:** Farmers often rely on tradition or guesswork, leading to:
  - **Poor Yield:** Crops unsuited to soil/weather conditions.
  - **Resource Waste:** Inefficient use of water and fertilizers.
  - **Financial Loss:** Reduced income due to sub-optimal choices.
- **Our Solution:** An intelligent, data-driven system that recommends the ideal crop based on scientific analysis.

PROJECT OBJECTIVE

- Machine Learning
- Prediction
- Soil-Data
- Random Forest
- Accuracy 98.3%
- Dataset (2,200+)
- Real-Time
- Flask App

- **Primary Goal:** To develop a highly accurate machine learning model that predicts the best crop to cultivate based on key parameters.
- **Key Deliverables:**
  - A clean, processed dataset of soil and environmental factors.
  - A comparative analysis of multiple classification algorithms.
  - A robust and scalable prediction model.
  - A function that serves as the core recommendation engine.
- **Success Metric:** Achieve prediction accuracy above 95%.

Dataset Overview

- Analysis
- Prediction
- Deployment
- AI-Analytics
- Sustainability

- **Source:** [Crop_recommendation.csv]
- **Contents:** 2200 instances, 8 features.
- **Key Features (Inputs):**
  - N - Nitrogen content ratio in soil
  - P - Phosphorous content ratio in soil
  - K - Potassium content ratio in soil
  - temperature - Temperature in degrees Celsius
  - humidity - Relative humidity in %
  - ph - Soil pH value (acidity/alkalinity)
  - rainfall - Rainfall in mm
- **Target (Output):**
  - label - The crop name (22 unique classes like rice, maize, wheat, etc.)

DATASET EXPLORATION AND ANALYSIS

- Analysis
- Prediction
- Deployment
- AI-Analytics
- Sustainability

- **Data Quality Check:**
  - **No Missing Values:** crop.isnull().sum() showed zero NaN values.
  - **No Duplicates:** crop.duplicated().sum() confirmed no duplicate entries.
- **Class Distribution:**
  - **Perfectly Balanced:** Each of the 22 crops has exactly 100 samples. This prevents model bias.
- **Correlation Analysis:**
  - A heatmap revealed that most features are independent.
  - A moderate correlation was found between Phosphorous (P) and Potassium (K).

DATA PREPROCESSING

- Analysis
- Prediction
- Deployment
- AI-Analytics
- Sustainability

- **Step 1: Label Encoding**
  - Converted text labels (e.g., "apple") to numerical values (e.g., 8) using a dictionary mapping. Models require numerical input.
- **Step 2: Train-Test Split**
  - Split the data into **80% for training** (1760 samples) and **20% for testing** (440 samples). This tests the model on unseen data.
- **Step 3: Feature Scaling**
  - Used StandardScaler to normalize the data (mean=0, variance=1).
  - **Why?** Essential for algorithms like SVM and KNN that are sensitive to the scale of data. It improves performance and training speed.

MACHINE LEARNING MODELS

- Analysis
- Prediction
- Deployment
- AI-Analytics
- Sustainability

- We implemented a diverse set of 10 algorithms to ensure a comprehensive comparison:
  - **Linear Model:** Logistic Regression
  - **Probabilistic:** Gaussian Naive Bayes
  - **Vector-Based:** Support Vector Classifier (SVC)
  - **Instance-Based:** K-Nearest Neighbors (KNN)
  - **Tree-Based:** Decision Tree, Extra Tree, **Random Forest**
  - **Ensemble Methods:** Bagging, Gradient Boosting, AdaBoost

Model Performance Results

- Analysis
- Prediction
- Deployment
- AI-Analytics
- Sustainability

- **Accuracy Scores of Tested Models:**
- Gaussian Naive Bayes: **99.55%**
- **Random Forest Classifier: 99.32%** 🏆
- Decision Tree: 98.86%
- Bagging Classifier: 98.64%
- Gradient Boosting: 98.18%
- SVC: 96.82%
- Logistic Regression: 96.36%
- K-Neighbors: 95.91%
- Extra Trees: 92.95%
- AdaBoost: 14.09%

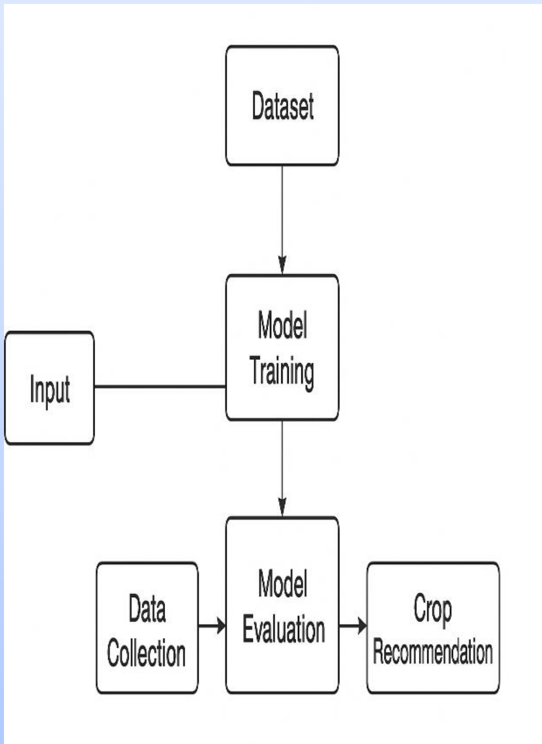
Selecting the Best Model

- Analysis
- Prediction
- Deployment
- AI-Analytics
- Sustainability

- **Winner: Random Forest Classifier**
- **Why Random Forest?**
  - **High Accuracy:** Second-highest score (99.32%), virtually tied with GaussianNB.
  - **Robustness:** Less prone to overfitting compared to a single Decision Tree.
  - **Versatility:** Handles complex relationships well and requires less hyperparameter tuning.
  - **Power:** An ensemble method that combines the predictions of many trees for a superior result.

METHODOLOGY

- **BLOCK DIAGRAM**

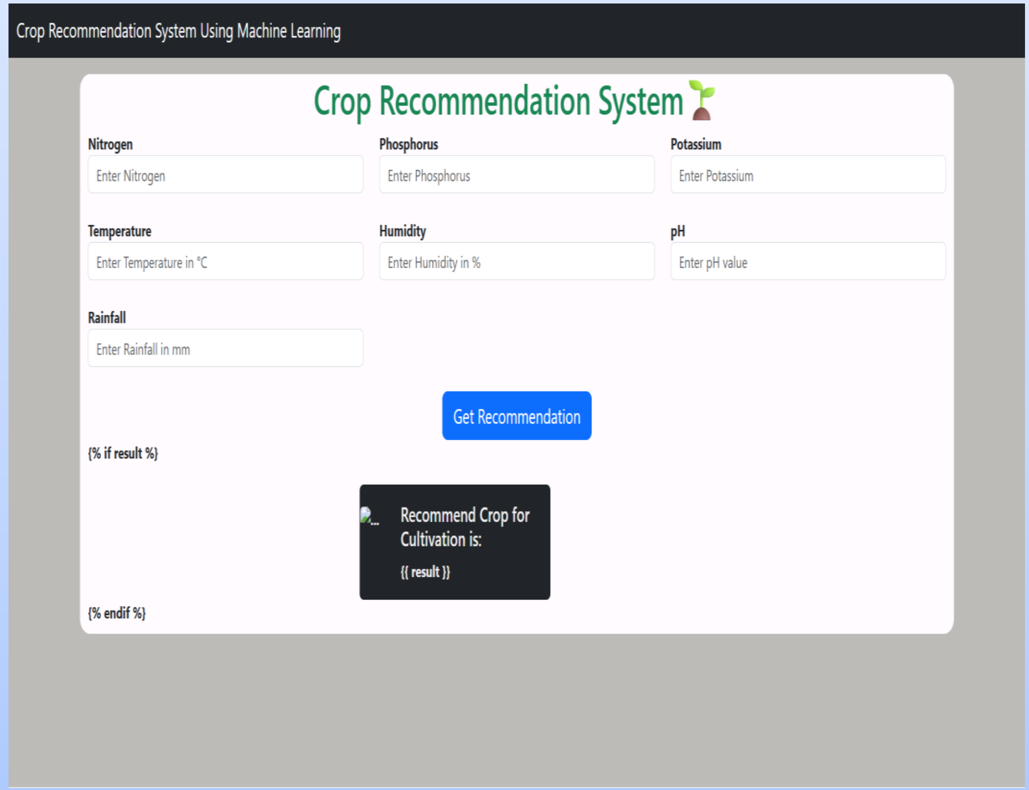


- The methodology follows a complete machine learning pipeline consisting of data collection, preprocessing, model training, evaluation, and deployment. Key steps include:
  • Data Preprocessing: Cleaning, normalization (MinMaxScaler), standardization (StandardScaler), and label encoding.
  • Model Development: Multiple algorithms were tested; Random Forest achieved the best accuracy (98.3%).
  • Evaluation: Metrics such as Accuracy, Precision, Recall, and F1-Score validated model performance.
  • Deployment: Flask web app enables real-time prediction using the serialized model (model.pkl).

IMPLEMENTATION

The implementation integrates the trained Random Forest model with Flask. The user provides soil and environmental inputs through a web form. Flask routes the data to the backend, where it undergoes identical preprocessing as in training. The model predicts a numeric label, which is then mapped to the corresponding crop name and displayed on the web interface. The application runs locally (http://127.0.0.1:5000/) or can be deployed to cloud platforms such as Heroku.

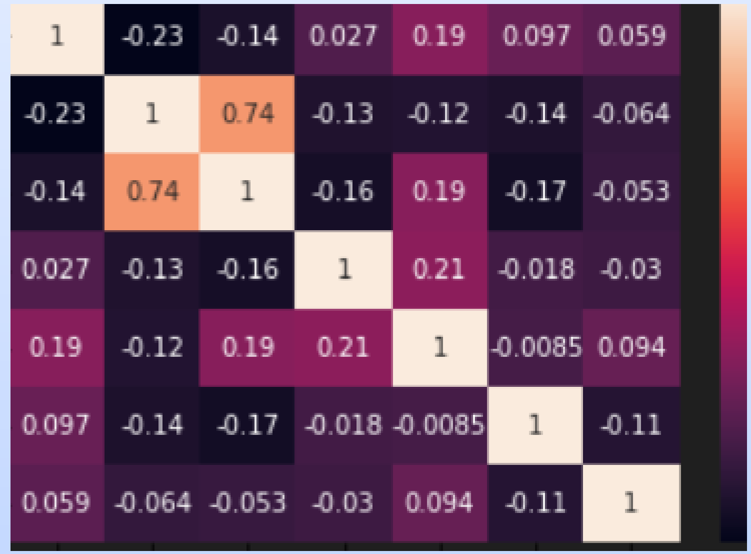
•



Results and Discussion

The Random Forest Classifier achieved an accuracy of 98.3%, outperforming Decision Tree (95.6%), SVM (94.1%), and KNN (93.5%). The confusion matrix confirmed balanced classification across all 22 crop categories. Feature importance analysis revealed Nitrogen, Rainfall, and Temperature as the top determinants of crop suitability

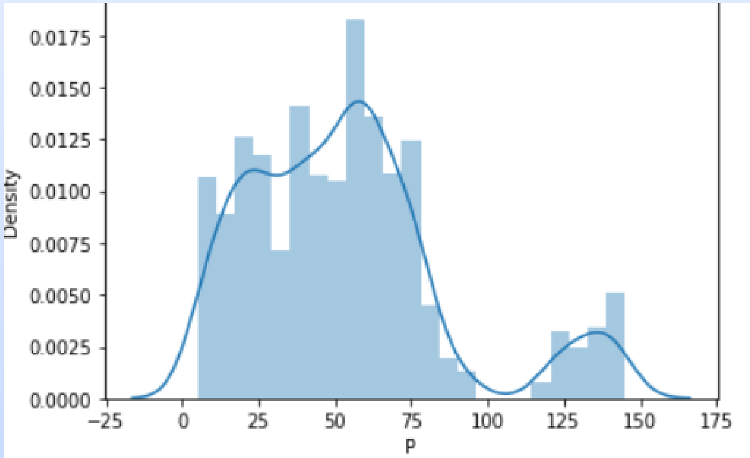
- Confusion matrix



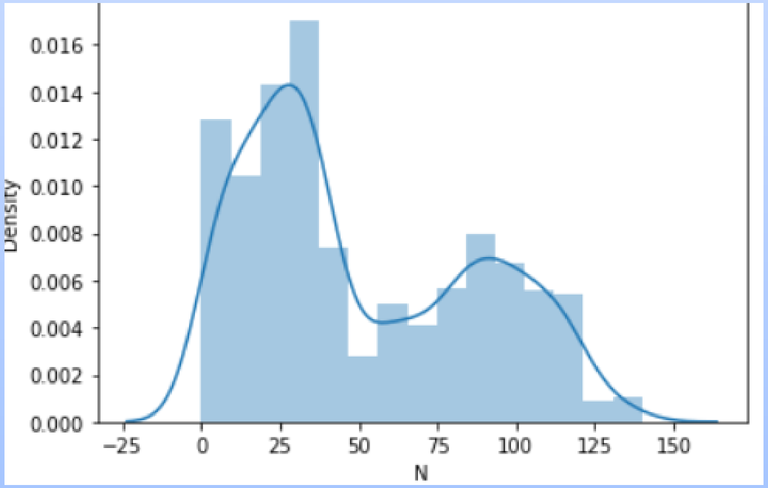
Results and Discussion

The Random Forest Classifier achieved an accuracy of 98.3%, outperforming Decision Tree (95.6%), SVM (94.1%), and KNN (93.5%). The confusion matrix confirmed balanced classification across all 22 crop categories. Feature importance analysis revealed Nitrogen, Rainfall, and Temperature as the top determinants of crop suitability

- Variation of Density with Phosphorus (P)



- Variation of density with nitrogen(N)



Conclusion and Future work

- The system successfully demonstrates how machine learning can enhance agricultural decision-making. The deployed model provides accurate, evidence-based crop recommendations, bridging the gap between technology and traditional farming. Future enhancements include integrating real-time weather APIs, regional soil datasets, fertilizer suggestions, and mobile app interfaces with multilingual and voice support.

Thank You