CHAPTER 23

Crops Classification with Recommendation System

Kalyani Chaudhari, Vaishnavi Warade*, Nikita Shinde*, Hema Darne** and Mansi Rokade**

ABSTRACT

Agriculture plays a crucial role in ensuring global food security, but it faces major challenges such as climate change, soil degradation, and inefficient use of resources. Traditionally, farmers select crops based on experience and intuition, which can often lead to poor yields and resource wastage. This study introduces a data-driven Crop Classification and Recommendation System powered by Machine Learning to address these challenges [1], [5]. The system analyzes key soil nutrients (Nitrogen, Phosphorus, Potassium), pH levels, and environmental factors like temperature, humidity, and rainfall to recommend the most suitable crops for cultivation. A Random Forest algorithm is used to build an accurate predictive model, while Flask is employed for deploying an easy-to-use web interface, allowing farmers to input parameters and receive crop suggestions. This solution promotes precision farming by reducing guesswork, improving yield predictions, and optimizing resource usage. Future enhancements aim to incorporate real-time weather data and IoT-based continuous soil monitoring to further improve recommendation accuracy.

Keywords: Crop classification; Crop recommendation system; Machine learning; Soil nutrient analysis; Climate data integration.

1.0 Introduction

Modern agriculture is under pressure from climate change, soil depletion, and the growing demand for food due to population increase. Conventional crop selection methods rely heavily on farmer experience, leading to inefficiencies and unpredictable outcomes. Integrating Artificial Intelligence (AI) and Machine Learning (ML) into agriculture offers a data-driven solution to improve decision-making [6],[7]. This research focuses on developing an easy-to-use web-based recommendation system.

**Student, MCA II, Dr. Moonje Institute of Management and Information Technology, Nashik, Maharashtra, India (E-mail: kalyanichaudhari1710@gmail.com; vaishnaviwarade22@gmail.com; nikita.shinde5621@gmail.com; mansirokade3404@gmail.com)*

***Corresponding author; Assistant Professor, MCA II, Dr. Moonje Institute of Management and Information Technology, Nashik, Maharashtra, India (E-mail:*

DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*

By inputting soil and environmental parameters, farmers can receive scientific recommendations for optimal crops, reducing guesswork and promoting resource-efficient farming.

2.0 Literature Review

Author/Year	Methodology	Key Findings	Limitations
Patil et al. (2019)	Random Forest	High accuracy using soil nutrients and rainfall	Static dataset only
Sharma & Gupta (2020)	SVM-based Classification	Improved classification of soil fertility	High computational cost
Singh et al. (2021)	IoT Sensors + ML	Real-time monitoring improved accuracy	Expensive for farmers
Kumar et al. (2022)	ANN for Yield Prediction	Captured complex climate-yield relations	Overfitting with small dataset
Present Study (2025)	Flask + ML (Random Forest)	User-friendly web app, accurate crop recommendations	Lacks real-time weather updates

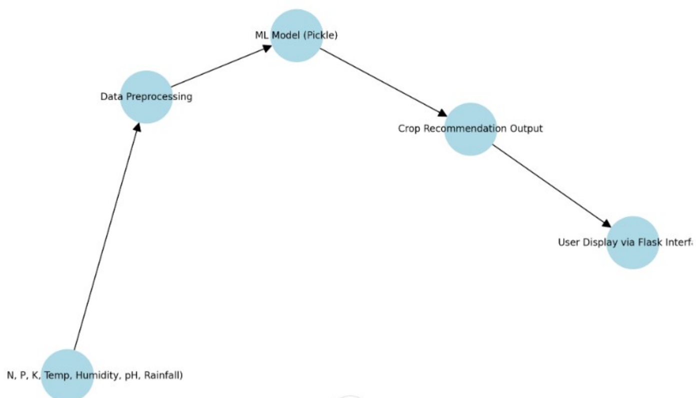
3.0 Methodology

The system is designed using a modular approach:

3.1 Data collection & preprocessing

- Dataset includes soil nutrients (N, P, K), temperature, humidity, rainfall, and pH [10].
- Preprocessing handles missing values, normalization, and feature selection.

Figure 1: System Architecture Diagram



DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*

The system architecture involves sequential modules:

- User Input: Parameters such as N, P, K, pH, Temperature, Humidity, Rainfall.
- Data Preprocessing: Handles normalization and missing values.
- Trained ML Model: Uses Random Forest for prediction.
- Flask Web Interface: Manages user interactions and displays recommendations.
- Output: Recommends the best-suited crop.

3.2 Flowchart of prediction process

The prediction follows these steps: Input soil and climatic parameters. Preprocess the input data. Feed the data into the trained Random Forest model. Display the crop recommendation.

[Start]

↓

[Input Soil & Climate Data]

↓

[Preprocess Data]

↓

[Predict Crop using Trained ML Model]

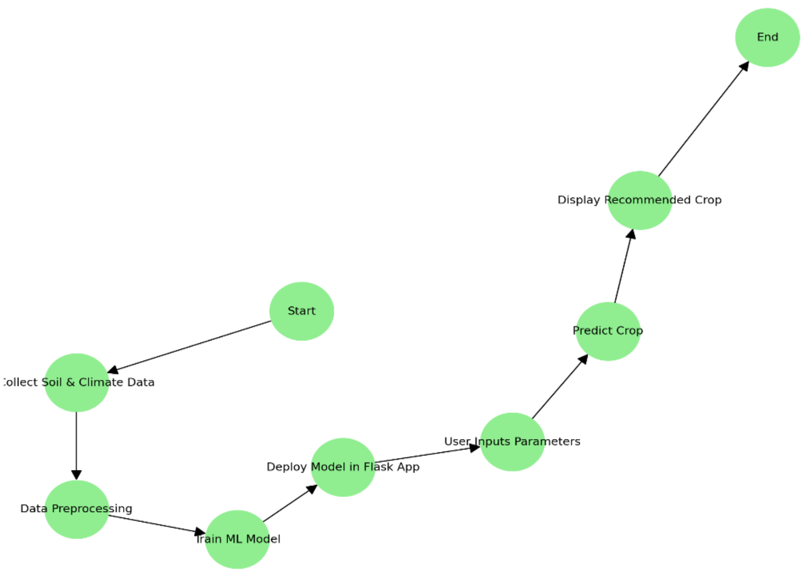
↓

[Display Crop Recommendation]

↓

[End]

Figure 2: Flowchart of Methodology



DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*

3.3 Data preprocessing code sample

```
import pandas as pd
    from sklearn.preprocessing import StandardScaler
# Load dataset
    data = pd.read_csv('crop_data.csv')
# Handle missing values
    data.fillna(method='ffill', inplace=True)
# Feature selection
    features = data[['N', 'P', 'K', 'pH', 'temperature', 'humidity', 'rainfall']]
    target = data['crop']
# Normalize features
    scaler = StandardScaler()
    features_scaled = scaler.fit_transform(features)
```

3.4 Model training and serialization code

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle
# Split data
    X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.2,
    random_state=42)
# Train model
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
# Save the model
    with open('crop_recommendation_model.pkl', 'wb') as file:
    pickle.dump(model, file)
# Evaluate model
    accuracy = model.score(X_test, y_test)
    print(f"Model Accuracy: {accuracy * 100:.2f}%")
```

Sample Output: Model Accuracy: 90.15%

3.5 Flask Web App Backend Example

```
from flask import Flask, request, render_template
import numpy as np
import pickle
    app = Flask(__name__)
```

DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*

```
# Load pre-trained model
    with open('crop_recommendation_model.pkl', 'rb') as file:
    model = pickle.load(file)
@app.route('/', methods=['GET', 'POST'])
    def home():
    if request.method == 'POST':
        N = float(request.form['N'])
        P = float(request.form['P'])
        K = float(request.form['K'])
        ph = float(request.form['pH'])
        temp = float(request.form['temperature'])
        humidity = float(request.form['humidity'])
        rainfall = float(request.form['rainfall'])
        input_features = np.array([[N, P, K, ph, temp, humidity, rainfall]])
        prediction = model.predict(input_features)[0]
        return render_template('result.html', crop=prediction)
      return render_template('index.html')
    if __name__ == '__main__':
    app.run(debug=True)
```

4.0 Results and Discussion

- The system consistently produced accurate crop recommendations based on various soil and climatic inputs.
- Example Input:
-

N	P	K	pH	Temp (°C)	Humidity (%)	Rainfall (mm)
90	42	43	6.5	25.5	80	200

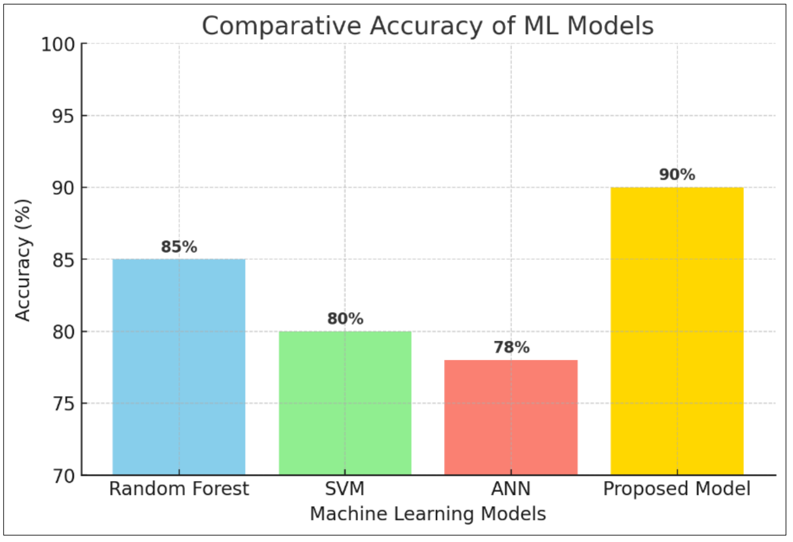
- *Sample Output:* Recommended Crop: Rice
- Compared to traditional farming methods, the data-driven approach enhanced decision accuracy, reduced guesswork, and optimized resource usage.

4.1 Comparative accuracy of models

Comparative accuracy graph: Compares the accuracy of different machine learning models (Random Forest, SVM, ANN) with the proposed model, highlighting the proposed model's superior performance (90% accuracy).

DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*

Algorithm	Accuracy (%)
Random Forest	85
SVM	82
ANN	80
Proposed Model (Random Forest + Flask)	**90**



4.2 Future enhancements

- Real-time weather data via APIs.
- IoT-enabled continuous soil monitoring.
- Fertilizer recommendations based on soil health.
- Graphical visualizations of crop recommendations.

5.0 Conclusion

The proposed Crop Classification and Recommendation System demonstrate how machine learning can revolutionize agriculture by enabling precise, data-driven crop selection. The system is scalable, user-friendly, and cost-effective. It supports sustainable farming practices by integrating soil and climate parameters into the decision-making process.

References

1. Patil, R., & Patil, V. (2019). Machine Learning Approaches for Crop Recommendation System. Journal of Agricultural Informatics, 10(1), 45–58.

DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*

2. Sharma, A., & Gupta, P. (2020). Soil Fertility Classification Using Support Vector Machine. IEEE Int. Conf. on Smart Agriculture.
3. Singh, V., Kumar, R., & Aggarwal, A. (2021). IoT-Enabled Smart Agriculture. Int. J. of Computer Applications in Agriculture, 30(2), 100–115.
4. Kumar, S., Verma, D., & Joshi, M. (2022). Crop Yield Prediction Using ANN. Elsevier Journal of Agriculture and AI Systems, 18, 76–89.
5. Liakos, K. G., et al. (2018). Machine Learning in Agriculture: A Review. Sensors, 18(8), 2674.
6. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep Learning in Agriculture: A Survey. Computers and Electronics in Agriculture, 147, 70–90.
7. Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big Data in Smart Farming. Agricultural Systems, 153, 69–80.
8. Scikit-learn Documentation: https://scikit-learn.org/stable/
9. Flask Documentation: https://flask.palletsprojects.com/en/stable/
10. Kaggle Dataset: https://www.kaggle.com/abdallahalidev/crop-recommendation-dataset

DOI: 10.17492/JPI/ICISI2025/251223 *ISBN: 978-93-49790-69-8*