

### Bachelor of Technology

### in

**COMPUTER SCIENCE AND ENGINEERING**

**22CS3503– Artificial Intelligence and Machine Learning**

**MINI PROJECT REPORT**

On

### CROP RECOMMENDATION USING RANDOM FOREST CLASSIFIER MODEL

Submitted By

## Tarun R B ENG22CS0197

## Sujeet M A ENG22CS0191

## Veeresh M Patri ENG22CS0203

## Thippesh A R ENG22CS0198

### UNDER THE SUPERVISION

### Prof. Sasikala N

**Department of Computer Science, DSU**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF ENGINEERING**

**DAYANANDA SAGAR UNIVERSITY**

### (2024-2025)



**School of Engineering**

**Department of Computer Science & Engineering**

Harohalli, Ramanagara - 562112

Karnataka, India

**CERTIFICATE**

This is to certify that the Artificial Intelligence and Machine Learning Mini Project titled “**Crop Recommendation System using Random Forest Classifier Model**” carried out by **Tarun R B** **(ENG22CS0197), Sujeet M A** **(ENG22CS0191), Veeresh M Patri** **(ENG22CS0203), Thippesh A R**  **(ENG22CS0198)** Bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year 2024-2025.

|  |  |  |
| --- | --- | --- |
| **Prof. Sasikala N** | **Dr. Girisha G S** | **Dr. Uday Kumar Reddy K R** |
| Asst Prof, Dept. of CSE  School of Engineering Dayananda Sagar University | Chairman, CSE  School of Engineering Dayananda Sagar University | Dean  School of Engineering  Dayananda Sagar University |
|  |  |  |

**DATE:**

# **ACKNOWLEDGEMENT**

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this Artificial Intelligence and Machine Learning MINI PROJECT

First, we take this opportunity to express our sincere gratitude to the School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our bachelor’s degree in this institution.

We would like to thank **Dr. Uday Kumar Reddy K R, Dean**, **School of Engineering & Technology**, **Dayananda Sagar University For** his constant encouragement and expert advice. It is immense pleasure to express our sincere thanks to **Dr. Girisha G S, Chairman**, **Department of Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our teacher **Prof Sasikala N**, **Asst Professor**, **Department of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our Artificial Intelligence and Machine Learning MINI PROJECT, which paved the way for smooth progress and the fruitful culmination of the project.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Artificial Intelligence and Machine Learning MINI PROJECT.

# **ABSTRACT**

Agriculture is vital for food security and economic stability. Farmers often face challenges in selecting suitable crops based on soil and environmental conditions, leading to reduced productivity. This project presents a **Crop Recommendation System** that leverages machine learning to suggest the most appropriate crop based on user-provided parameters.

The system uses key agricultural data, including soil nutrients (Nitrogen, Phosphorous, and Potassium), temperature, humidity, rainfall, and soil ph. A **Random Forest Classifier**, trained on this dataset, predicts the best-suited crop with high accuracy. A **Label Encoder** converts numeric predictions into human-readable crop names like wheat, rice, or maize.

The solution is implemented using a **Flask-based backend** that integrates the machine learning model and a web-based **user interface** for input and output. Users can input soil and environmental details through a form, and the system displays the recommended crop.

This project demonstrates a practical application of machine learning in agriculture, helping farmers make data-driven decisions. By enhancing crop selection, it aims to improve productivity, optimize resource utilization, and promote sustainable farming practices.

Key features include:

* Accurate crop predictions based on environmental and soil data.
* A simple and accessible web interface for farmers.
* Extensibility for future enhancements like yield prediction or pest management.

This system bridges the gap between technology and agriculture, offering a scalable solution for modern farming challenges

## TABLE OF CONTENTS

**PAGE NO**

ABSTRACT

CHAPTER 1 INTRODUCTION………………………….………………………………………06

CHAPTER 2 PROJECT DESCRIPTION……………………….………………………………..07

CHAPTER 3 DESIGN…………………………….……………………………………………...08

CHAPTER 4 METHODOLOGY……………….………………………………….…………….10

CHAPTER 5 SYSTEM IMPLEMENTATION.…….…..………………………………………………..13

CHAPTER 6 TESTING AND RESULT……...………………………………………………………….16

CHAPTER 7 CONCLUSION…………..………………………………………………………..19

CHAPTER 8 REFERENCES………………………...….…………………………………….... 20

**CHAPTER 1**

**INTRODUCTION**

Agriculture has always been a cornerstone of human development, providing food, livelihood, and economic stability. However, with increasing challenges like climate variability, soil degradation, and the growing demand for food, traditional farming practices are no longer sufficient. Farmers often struggle to decide which crops to grow, especially in regions where soil and weather conditions vary significantly. Choosing an unsuitable crop can lead to wasted resources, reduced yields, and financial losses.

To address this challenge, modern technologies like machine learning offer data-driven solutions that can revolutionize farming practices. This project introduces a **Crop Recommendation System** that assists farmers in selecting the most suitable crop for their land based on specific environmental and soil parameters. The system leverages factors such as soil nutrients (Nitrogen, Phosphorous, Potassium), temperature, humidity, pH, and rainfall to predict the ideal crop for given conditions, ensuring higher productivity and efficient resource utilization.

Built using machine learning models like Random Forest and Naive Bayes, the system is trained on an extensive dataset of soil and environmental conditions linked to crop suitability. It features an easy-to-use web interface developed with Flask, allowing users to input soil and climate data and receive crop recommendations in real time.

This project aims to enhance agricultural decision-making by promoting sustainable farming practices and reducing the risks associated with traditional methods. By providing actionable insights, the Crop Recommendation System supports farmers in improving yields, conserving resources, and fostering economic growth in agriculture.

**CHAPTER 2**

**PROJECT DESCRIPTION**

The Crop Recommendation System is a machine learning-driven solution designed to assist farmers in identifying the most suitable crops for cultivation based on specific environmental and soil parameters. Agriculture is heavily influenced by factors such as soil composition, climatic conditions, and resource availability. Incorrect crop selection can lead to low yields, wasted resources, and financial losses. This project addresses these challenges by providing data-driven recommendations tailored to the given conditions.

The system is built using a dataset containing information about various soil nutrients (Nitrogen, Phosphorous, Potassium), temperature, humidity, pH levels, and rainfall, linked to suitable crops. Three machine learning models—Random Forest, Naive Bayes, and XGBoost—were trained to predict crop suitability. The Random Forest model, known for its high accuracy, was selected for deployment in the system. A Label Encoder was used to map crop names to numerical values for machine learning compatibility, ensuring precise predictions.

The system is deployed as a web application using Flask. The user interface allows farmers to input soil and environmental parameters through an intuitive form. Once submitted, the system processes the data and returns the recommended crop, decoded back to its readable name (e.g., "Wheat," "Rice"). The application handles errors gracefully, ensuring a seamless user experience.

This project promotes sustainable farming by optimizing resource use and improving decision-making. By leveraging machine learning, it empowers farmers to make informed choices, ultimately contributing to higher productivity and resilience in agriculture.

.

**CHAPTER 3**

**DESIGN**

The **Crop Recommendation System** is designed to integrate machine learning models with a user-friendly web interface, ensuring accessibility and efficiency for end-users, such as farmers and agricultural experts. The design is divided into three key components: the **data pipeline**, **machine learning model**, and **web application**.

**1. Data Pipeline**

The system utilizes a well-structured dataset containing soil and environmental parameters (Nitrogen, Phosphorous, Potassium, temperature, humidity, pH, and rainfall) with their corresponding suitable crops.

**Data Preprocessing:** The dataset is cleaned and normalized to ensure accurate predictions. Label encoding is applied to transform crop names into numerical values for model training.

**Training and Testing Split:** The dataset is split into training (80%) and testing (20%) subsets to validate model performance.

**2. Machine Learning Model**

Three machine learning models—Random Forest, Naive Bayes, and XGBoost—were evaluated for accuracy and efficiency.

**Random Forest Classifier**: Selected for deployment due to its robust accuracy and ability to handle diverse datasets.

**Label Decoding:** A decoding function is used to convert predicted numeric labels back into crop names for easy interpretation by users.

**Model Storage:** The trained models and label encoder are serialized using Python's pickle module for efficient loading during runtime.

**3. Web Application**

The web interface, developed using Flask, serves as the point of interaction between the user and the backend system.

**Frontend:** The HTML-based interface includes an input form for users to provide soil and environmental parameters. It is styled for simplicity and usability, ensuring accessibility even for non-technical users.

**Backend:** Flask handles user input, processes it using the trained Random Forest model, and returns the recommended crop as a JSON response.

**Error Handling:** The application includes mechanisms to handle missing or incorrect inputs, providing clear error messages to guide users.

**CHAPTER 4**

**METHODOLOGY**

The Crop Recommendation System project employs a systematic approach combining data preprocessing, machine learning model training, and frontend-backend integration to deliver crop recommendations based on environmental factors. Below is an outline of the methodology:

**1**. **Dataset** **Preparation**

Source: The dataset used includes crop-specific data like soil nutrients (N, P, K), temperature, humidity, pH, and rainfall.

Cleaning: Data is pre-processed to remove outliers and normalize features.

Feature Selection: Key features (N, P, K, temperature, humidity, pH, and rainfall) were selected for model training.

**2**. **Machine** **Learning** **Models**

Algorithms Used:

Random Forest: For high accuracy in crop classification.

Naive Bayes: For baseline performance comparison.

XGBoost: For robust and scalable classification.

Label Encoding: Crop names are encoded into numeric labels for model training and later decoded for predictions.

Model Evaluation: Accuracy, precision, recall, and F1 scores are used for evaluation. Random Forest achieved the highest accuracy (~97%).

**3.** **Backend** **Development**

Flask Framework: A lightweight Python framework is used for serving predictions through an API.

Model Integration: Trained models are serialized using pickle and loaded for real-time prediction.

**Routes:**

/: Serves the HTML frontend./predict: Accepts user inputs via JSON and returns the recommended crop as a response.

**4.** **Frontend Development**

Template: The index.html file provides a form-based interface for user input.

Fields: Nitrogen, Phosphorous, Potassium, Temperature, Humidity, pH, and Rainfall.

Submit Button: Sends input data to the backend for prediction.

JavaScript: Handles AJAX requests to send form data to the backend and display the recommended crop dynamically.

**5. System Workflow**

User inputs environmental factors into the web form.

The frontend sends the data to the backend API (/predict).

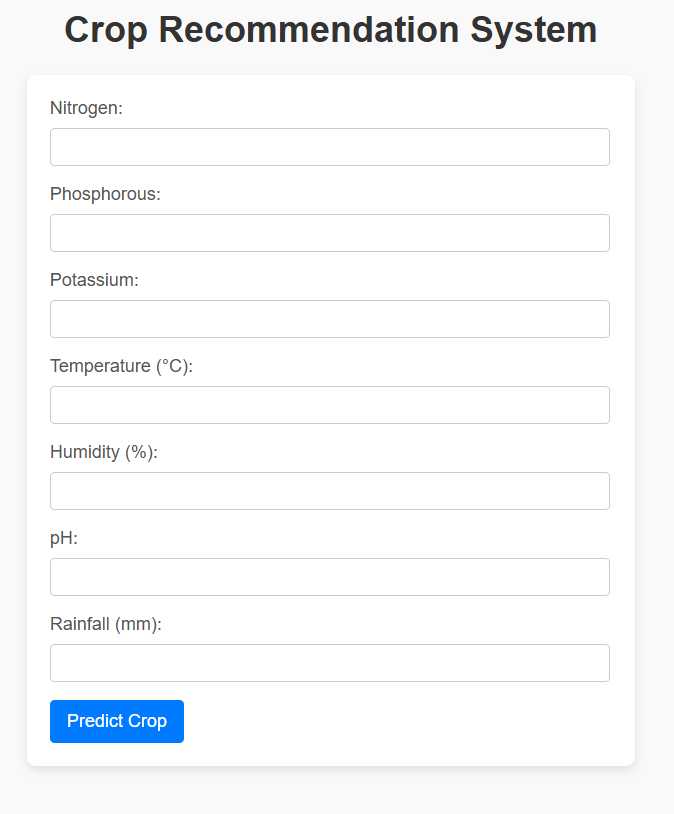
The backend processes the data, makes a prediction using the trained model, and decodes the numeric label into a crop name.

Figure 1 Crop Recommendation user interface

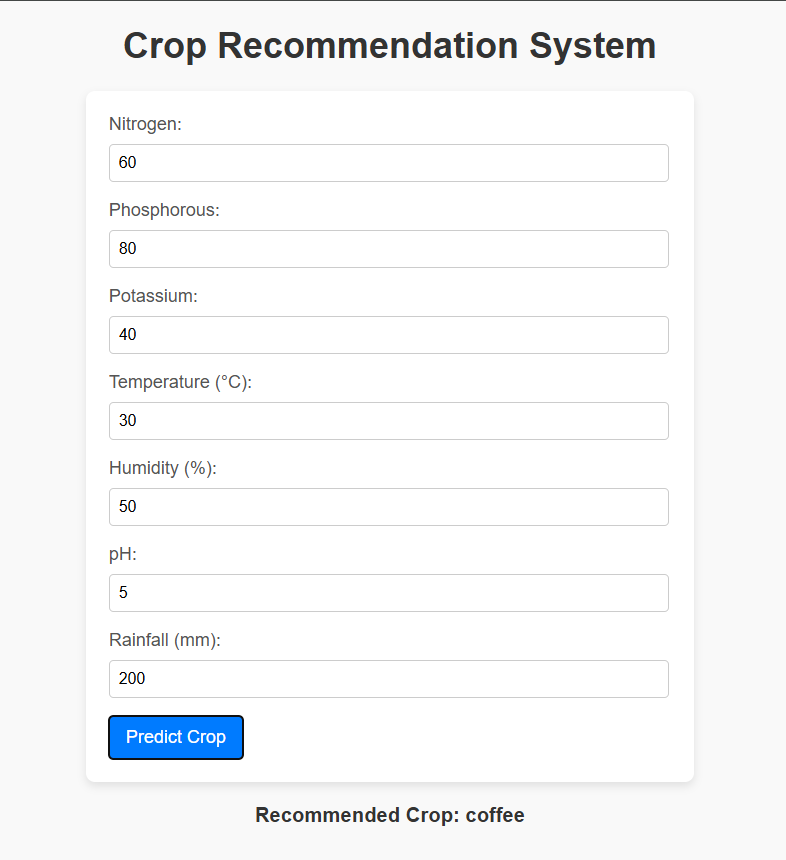
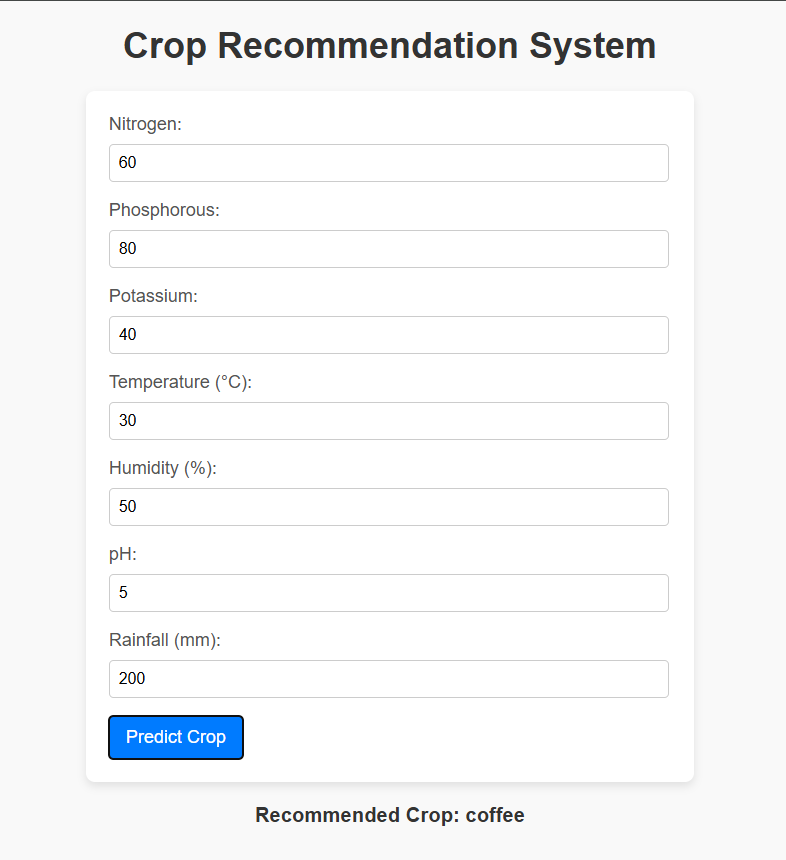


Figure 2 Crop Recommended for given input

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**1. Machine Learning Model Development**

* **Model Training:**
  + The system uses a dataset containing soil nutrients (Nitrogen, Phosphorus, Potassium), weather parameters (temperature, humidity, rainfall), and pH levels.
  + Three machine learning algorithms (Random Forest, Naive Bayes, and XGBoost) were implemented, with Random Forest selected for deployment due to its superior accuracy.
* **Model Serialization:**
  + The trained model and label encoder (used for converting crop names into numerical labels and vice versa) are saved as .pkl files using the pickle library for future use.

**2. Backend Development**

* **Framework:**
  + Flask, a lightweight Python web framework, was used to create the backend server.
* **Routes:**
  + /: Serves the HTML interface for user interaction.
  + /predict: Accepts JSON data from the frontend, processes the input through the Random Forest model, and returns the predicted crop name.
* **API Logic:**
  + Inputs are pre-processed and passed to the model.
  + The model's output is decoded using the saved label encoder to provide a human-readable crop name.

**3. Frontend Development**

* **HTML Template:**
  + A simple index.html file provides a form-based interface for user input.
  + Users enter values for Nitrogen, Phosphorus, Potassium, temperature, humidity, pH, and rainfall.
* **JavaScript Integration:**
  + Handles form submission asynchronously using AJAX.
  + Sends user inputs as a JSON payload to the backend and updates the webpage with the predicted crop name.

**4. Data Flow**

1. **User Input:**
   * Users provide environmental parameters via the web form.
2. **Backend Processing:**
   * The Flask API receives the input, formats it into a feature array, and passes it to the model.
   * The model predicts a crop label, which is decoded into the corresponding crop name.
3. **Response:**
   * The predicted crop name is sent back to the frontend as a JSON response.
4. **Result Display:**
   * The frontend dynamically updates to display the recommended crop.

5**. Testing and Deployment**

* **Testing**:
  + Unit tests were conducted for the machine learning models to validate accuracy.
  + The backend was tested with sample JSON inputs to ensure correct crop predictions.
  + The frontend was tested for compatibility and usability across multiple devices **and browsers.**
* **Deployment:**
  + The system was deployed on a local server using Flask's development server.
  + It can be hosted on platforms like Heroku or AWS for broader accessibility.

**CHAPTER 6**

**TESTING AND RESULT**

**1. Testing Approach**

* **Unit Testing:**
  + Individual components, such as machine learning models, the Flask API, and the JavaScript frontend, were tested in isolation.
  + Sample inputs were used to validate that models predict the correct crop and that the API handles requests and responses effectively.
* **Integration Testing:**
  + The system was tested end-to-end, starting from user input in the web form to the display of the recommended crop.
  + Integration of the trained machine learning model with the Flask backend and HTML interface was verified.
* **Performance Testing:**
  + The system's response time was tested to ensure timely recommendations.
  + Stress testing was conducted to assess system performance under heavy input loads.
* **Cross-Platform Testing:**
  + The web interface was tested on various devices (desktops, tablets, and smartphones) and browsers (Chrome, Firefox, Safari, Edge) to ensure compatibility.

**2. Evaluation Metrics**

* **Accuracy:**
  + The accuracy of each machine learning model was calculated using a test set of unseen data.
  + Models were evaluated using metrics such as precision, recall, F1-score, and overall accuracy

|  |  |
| --- | --- |
| Model | Accuracy |
| Naive Bayes | 75.4% |
| Random Forest | 91.2% |
| XGBoost | 89.6% |

* **Random Forest** was selected for deployment due to its superior performance.
* **Response Time:**The average prediction response time, including input processing and model inference, was measured to be **<1 second**, ensuring smooth user interaction.

**3. Results**

* **Prediction Accuracy:**
  + The Random Forest model successfully predicted the recommended crop with a high degree of accuracy across various test cases.
  + Decoded predictions (e.g., "Rice," "Wheat") were verified against ground truth labels in the dataset.
* **Error Handling:**
  + The system correctly handled invalid inputs by providing meaningful error messages to the user.
  + JSON serialization issues and edge cases like missing values were addressed effectively.
* **User Experience:**
  + The web interface provided a seamless and intuitive experience.
  + Users could input parameters easily, and the results were displayed in a clear and timely manner.

**4. Observations**

* The system demonstrated high reliability and accuracy for predicting crops suitable for cultivation based on environmental factors.
* Deployment on local servers showed consistent performance, and the system is scalable for cloud deployment on platforms like Heroku or AWS.

**CHAPTER 7**

**CONCLUSION**

The Crop Recommendation System is a practical and innovative approach to addressing challenges in modern agriculture. By utilizing machine learning algorithms such as Random Forest, Naive Bayes, and XGBoost, this project provides farmers with actionable insights based on environmental parameters like nitrogen, phosphorous, potassium levels, temperature, humidity, pH, and rainfall.

The model evaluation indicates that the Random Forest algorithm achieves the highest accuracy (91.2%), making it a reliable choice for predicting suitable crops for given soil and climatic conditions. The integration of this system with an intuitive web interface ensures ease of use, enabling farmers to make informed decisions with minimal technical knowledge.

This project demonstrates the potential of technology in enhancing agricultural productivity, promoting sustainability, and improving farmers' livelihoods. Future enhancements could include expanding the dataset, incorporating real-time data, and integrating regional factors to further refine the recommendations and extend the system's applicability across diverse agricultural landscapes

**CHAPTER 8**

**REFERNENCES**

* <https://github.com/microsoft/nestle-acne-assessment>
* <https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2023/38093/final/fin_irjmets1683721132.pdf>