A PROJECT REPORT
ON

"CROP RECOMMENDATION SYSTEM"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELORS' DEGREE IN
COMPUTER SCIENCE & ENGINEERING
BY

SULAV BHANDARI	21053470
RANJAN SRIVASTAV	21053402
SAPHAL PANTH	21053458
CHANDRA B. CHHETRI	21053415

UNDER THE GUIDANCE OF
DR. ARGHYA KUNDU

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA – 751024

APRIL 2024

KIIT Deemed to be University
School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the course entitled

"Crop Recommendation System"

submitted by

SULAV BHANDARI	21053470
RANJAN SRIVASTAV	21053402
CHANDRA BAHADUR CHHETRI	21053415
SAPHAL PANTH	21053458

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be University, Bhubaneswar. This work is done during year 2024.

Date: 14/11/2024

Dr. ARGHYA KUNDU

(Guide Name)
Project Guide

Acknowledgements

We extend gratitude to the agricultural community, researchers, and organizations that provided invaluable data and insights. Their contributions were instrumental in developing this system, which aims to make a meaningful impact on sustainable farming practices.

We are also profoundly grateful to **Dr. ARGHYA KUNDU** of **Kalinga Institute of Industrial Technology** for his time and support in evaluating our project work. His insights and guidance are greatly appreciated, and his feedback will undoubtedly contribute to my continued learning and professional development.

SULAV BHANDARI - 21053470
RANJAN SRIVASTAV - 21053402
CHANDRA BAHADUR CHHETRI - 21053415
SAPHAL PANTH - 21053458

ABSTRACT

This is basically a crop recommendation system which uses machine learning application for farmers and the profession involved in agriculture to select their optimal crops based on environmental and soil parameters. The system has considered most critical factors in soil type, climate, temperature, rainfall, humidity, and pH levels to give specific crop recommendations tailored to unique geographic locations. The application integrates several machine learning algorithms such as Decision Trees, Random Forests, Support Vector Machine (SVM), and Gradient Boosting for the pre-processing of historical data related to agriculture to predict appropriate crops to be cultivated within a particular area in terms of yield potential and economic returns.

Data Preprocessing has been applied to this project to ensure quality data used to train the models. The python-based libraries, namely Scikit-learn, Pandas, and NumPy, have been used to train, evaluate, and predict the model. A user-friendly interface developed with Flask and HTML/CSS will allow users to input relevant data easily, and based on it, give the users tailored recommendations. Integration into sustainable agricultural practices can help eliminate the risks associated with inappropriate crop choices. Future updates include the integration of real-time weather data, profitability analysis, and a mobile application for ultimate accessibility. This is an ideal tool for facilitating decision-making by using data in agriculture, thus ensuring better crop management and enhancement of productivity.

Contents

Chapter 1

Introduction

Agriculture is key in the global economic system as it provides food, raw materials, and employment for the billions. The problems farmers face in terms of crop selection arise from the differences in soil types, uncertain climate conditions, and different environmental conditions. Wrong crop selection for a particular area or season may lead to the reduction of yields, financial loss, and wastage of resources. The innovations of data science and machine learning addressed these issues. Due to these advancements in technology, new data-driven agricultural methods are now possible in farming. The Crop Recommendation System is based on the application of machine learning algorithms; determining some of the major input parameters related to crops-that is, properties of soils such as nitrogen, phosphorus, and potassium along with pH along with climate data, including temperature, rainfall, and humidity. It will offer region-specific crop suggestions to the system, whereby the yield and profitability of producers will increase.

The project is very accurate and adaptive with different environments using techniques like Decision Trees, Random Forests, SVM, and Gradient Boosting made from machine learning. This dataset provides historical crop and environmental data, through which the system identifies intricate variable relationships. It provides easy data entry with real-time recommendations using an interactive web interface. Innovation helps reduce crop failure risks while boosting agricultural productivity and sustainability. The system modernizes agriculture, providing farmers with reliable data-driven guidance in crop choice in the face of changing climate and market conditions.

Chapter 2

Basic Concepts/ Literature Review

This chapter discusses the theoretical basis and tools in developing the Crop Recommendation System. It leads to some of the most vital concepts, models, and web-development frameworks of the project.

2.1 Machine Learning Models

Crop Recommendation System is based on machine learning: supporting data-driven prediction depending upon soil and environment parameters.

Random Forest Classifier

- **Definition:**
  Random Forest is an ensemble learning method that makes hundreds of decision trees during training and outputs the mode of their predictions (classification) or mean (regression).

- **Why Random Forest?**
  - Handles categorical and numerical data well.
  - Avoids overfitting by averaging many decision trees.
  - Resilient to missing and noisy data.
  - Excellent performance with non-linear datasets.

- **Application in This Project:**
  Random Forest was selected as the main model because it performed outstandingly during both the training and testing phases. It would correctly identify the crops given many inputs like Nitrogen, Phosphorus, and Temperature with an improved accuracy.

Gaussian Naive Bayes (GNB)

- **Definition:**
  A probabilistic classifier which applies Bayes' theorem since its features are supposed to be independent from each other.

- **Why Gaussian Naive Bayes?**
  - It's fast and computationally efficient
  - Used for problems whose data is normally distributed.
  - Useful as a baseline model for comparison.

- **Application in This Project**
  GNB was used to provide benchmarking with good performance and minimal computation.

Other Models Used

- **Support Vector Machine (SVM):**
  Accurate but computationally expensive with large data size.

- **K-Nearest Neighbors (KNN):**
  Computational and time intensive, required a lot of extensive tuning of parameters, not efficient with large feature sets.

- **Gradient Boosting and AdaBoost:**
  Had good performance but were slower than Random Forest.

The models were evaluated on the basis of accuracy metrics as well as confusion matrices, after which the best algorithm found was Random Forest.

2.2 Data Preprocessing Tools

Data pre-processing is very vital in the consistency and improvement in model performance. Techniques and tools that were used are:

MinMaxScaler

- **Purpose:**
  It scales each feature using minimum and maximum values it has into a range of 0 to 1 that transforms the data.

- **Application in this project:**
  The features like Nitrogen, Temperature, and Rainfall had very different ranges. MinMaxScaler normalized such that no feature was dominant in the model's consideration.

Label Encoding

- **Purpose:**
  Converts categorical labels, such as crop names, into a number.

- **Application in This Project:**
  A dictionary called “crop_dict” is simply created to map crop names to integers so that machine learning algorithms can process the data.

2.3 Flask Framework and Web Development Tools

Using the Flask framework along with other state-of-the-art web development tools, it aims to provide an interface toward this system.

Flask

- **Definition:**
  A simple, flexible and lightweight web framework for Python as simple as it is flexible.

- **Features of Flask:**
    - Modular design allows seamless integration of multiple components.
    - Built-in development server for testing and debugging.
    - Jinja2 templating engine for rendering dynamic HTML content.

- **Application in This Project:**
  Flask serves as the backend, handling:
    - Routing requests (e.g., / for the homepage and /predict for predictions).
    - Rendering the web interface (index.html).
    - Passing user inputs to the trained Random Forest model and displaying results dynamically.

Frontend Tools

The frontend interface ensures accessibility and ease of use for the end-users.

Bootstrap

- **Purpose:**
  A popular CSS framework that simplifies responsive web design.

- **Application in This Project:**
    - Navbar, form layouts, and buttons were styled using Bootstrap classes.
    - Ensured the application was mobile-friendly and visually appealing.

Custom CSS

- **Purpose:**
  Custom styling was applied to align the interface with a modern aesthetic.

- **Highlights from styles.css:**
  - **Background Color:** Light gray (#f7fafc) for a clean, professional look.
  - **Primary Color:** Green (#38a169) for buttons and headings, symbolizing agriculture.
  - **Hover Effects:** Buttons enlarge slightly on hover, improving user interaction.

2.4 Literature Review

Related Work:
Several machine learning systems have been developed for agriculture. Key findings include:

- **Existing Gap**: Most systems are limited to single-region datasets or fail to integrate real-time user inputs.
- **Innovative Solution**: The **Crop Recommendation System** improves upon these limitations by using a scalable model and a responsive web interface.

Conclusion
This chapter provides the necessary foundation for understanding the tools and techniques that enable the seamless functioning of the **Crop Recommendation System**. These technologies collectively enhance the project's usability and effectiveness, bridging the gap between machine learning and agriculture.

Chapter 3

Problem Statement / Requirement Specifications

This chapter introduces the general problem to be addressed by the **Crop Recommendation System** as well as the procedure to delineate and establish requirements, planning, and design of the system.

3.1 Problem Statement

3.1.1 The Problem

Food production is still agriculture mainstream worldwide, and the sad reality is that most farmers decide which crops to grow solely on intuition or based on minimal knowledge. This inevitably leads to poor crop selections and wasted resources-more so when the environmental pressures are changing rapidly.

3.1.2 Challenges

1. **Insufficient Data-Driven Insights:**
   Farmers require tailored recommendations that take multiple variables such as soil properties and environmental factors into account.

2. **Environmental Variability:**
   Unpredictable weather patterns create uncertainty in crop planning and management.

3. **Efficient Resource Utilization:**
   It's critical to maximize the effective use of fertilizers, water, and land for better sustainability.

4. **Ease of Use:**
   Tools designed to aid farmers should be simple, intuitive, and widely accessible, particularly in rural areas with varying levels of digital literacy.

3.1.3 The Solution

The **Crop Recommendation System** leverages machine learning to analyze soil and environmental data, providing accurate crop suggestions to farmers. By integrating data science with a user-friendly web platform, the system bridges the gap between technology and practical agricultural needs.

3.2 Requirement Specifications

The specifications for this project were drawn up following a structured approach that ensures both functionality and usability.

3.2.1 Functional Requirements

1. **User Inputs:**
    - Accept soil properties like Nitrogen, Phosphorus, and Potassium.
    - Capture environmental factors, including Temperature, Humidity, pH, and Rainfall.

2. **System Processing:**
    - Normalize user-provided data using MinMaxScaler for consistency.
    - Use a pre-trained Random Forest model to predict the best crop for the given conditions.

3. **Outputs:**
    - Provide a clear and concise recommendation, such as, "Based on the provided inputs, Maize is the best crop for cultivation."

4. **Error Management:**
    - Ensure validation of inputs (e.g., no empty fields or invalid values).
    - Display helpful error messages if the system cannot make a recommendation.

3.2.2 Non-Functional Requirements

1. **User Accessibility:**
    - The platform should be simple enough for non-technical users, with mobile compatibility ensured through responsive design.

2. **Performance:**
    - The recommendation process, including prediction and response, must be swift—ideally under two seconds.

3. **Scalability:**
    - The system should easily adapt to include new crops or additional environmental parameters as data availability grows.

4. **Reliability:**
    - Deliver accurate predictions, minimizing errors in crop recommendations.

5. **Data Security:**
    - Safeguard input data from unauthorized access or misuse.

3.3 System Design

3.3.1 Design Constraints

1. **Hardware Setup**
    - **Server:** Moderate computational resources to host the Flask-based application.
    - **Client:** A device with any modern browser, such as a smartphone, tablet, or desktop.

2. **Software Requirements:**
    - **Backend:** Flask for server-side processing.
    - **Frontend:** HTML, CSS, and Bootstrap for an intuitive and visually appealing interface.

    Machine Learning: Libraries like Scikit-learn to train, test, and serialize the prediction model.

3. **Development Environment:**
    - The model was trained using a dataset of 22 crops, focusing on their growth requirements and environmental needs. Features such as Nitrogen, Rainfall, and pH were standardized to fit within a uniform scale for better model performance.

3.3.2 System Architecture

The system architecture integrates three distinct layers:

1. **Input and Preprocessing Layer**
    - Accepts soil and environmental parameters entered by users.
    - Prepares the data for prediction by normalizing inputs using MinMaxScaler.

2. **Prediction Layer**
    - Utilizes a serialized Random Forest model to process preprocessed data and determine the most suitable crop.

3. **Output Layer**
    - Dynamically displays results on the web interface, providing users with clear and actionable crop recommendations.

3.4 Project Planning

3.4.1 Steps Followed

1. **Requirement Analysis:**
    - Investigated critical parameters affecting crop growth, including soil nutrients and weather conditions.
    - Studied and refined the dataset to ensure it covered diverse agricultural scenarios.

2. **Model Development:**
    - Preprocessed data to make it machine-learning-ready.
    - Trained and tested multiple algorithms (e.g., Logistic Regression, Random Forest) to identify the best-performing model.

3. **Interface Design:**
    - Designed a responsive and intuitive web interface, leveraging Bootstrap for mobile compatibility and clean layouts.

4. **Testing and Validation:**
    - Conducted extensive testing to ensure that the system provided accurate recommendations and handled invalid inputs gracefully.

The **Crop Recommendation System** addresses a pressing issue in agriculture by combining the precision of machine learning with the accessibility of web-based tools. Through careful planning and design, the system provides farmers with reliable, real-time crop recommendations, improving decision-making and resource management.

Chapter 4

Implementation

This chapter explains how the **Crop Recommendation System** was born. Every step involved from data preparation to model development is explained in detail and covers aspects of system testing and result validation besides quality measures taken for ensuring the system.

4.1 Methodology

The implementation followed a systematic approach, ensuring all technical requirements were addressed effectively.

4.1.1 Data Preprocessing

Before training the machine learning models, the dataset was processed to make it consistent and suitable for analysis:

1. **Dataset Overview:**
    - The project was based upon the Crop_recommendation.csv file which itself provides the properties of soil (Nitrogen, Phosphorus, Potassium) and environmental factors such as temperature, humidity, pH and rainfall for 22 different crops.
    - Ensured the dataset was free of missing values and duplicates.

2. **Encoding Crop Labels:**
    - Since machine learning models work with numerical data, the crop names (e.g., Rice, Maize) were converted into numbers using a mapping dictionary.

3. **Feature Scaling:**
    - To standardize the input features (e.g., Nitrogen, Temperature), the MinMaxScaler was applied, scaling all values to fall within the range of 0 to 1.
    - This prevented any single feature from dominating the prediction process.

4.1.2 Training Machine Learning Models

Several machine learning algorithms were evaluated to find the best fit for the project.

Model	Key Features
Logistic Regression	Simple and interpretable, but less effective for complex relationships.
Random Forest Classifier	Combines multiple decision trees to improve accuracy and reduce overfitting.
Gaussian Naive Bayes	Fast and efficient, though not as accurate for this dataset.
Gradient Boosting	Improved prediction accuracy but required more computational resources.
K-Nearest Neighbors	Worked well with parameter tuning but was computationally intensive for testing.

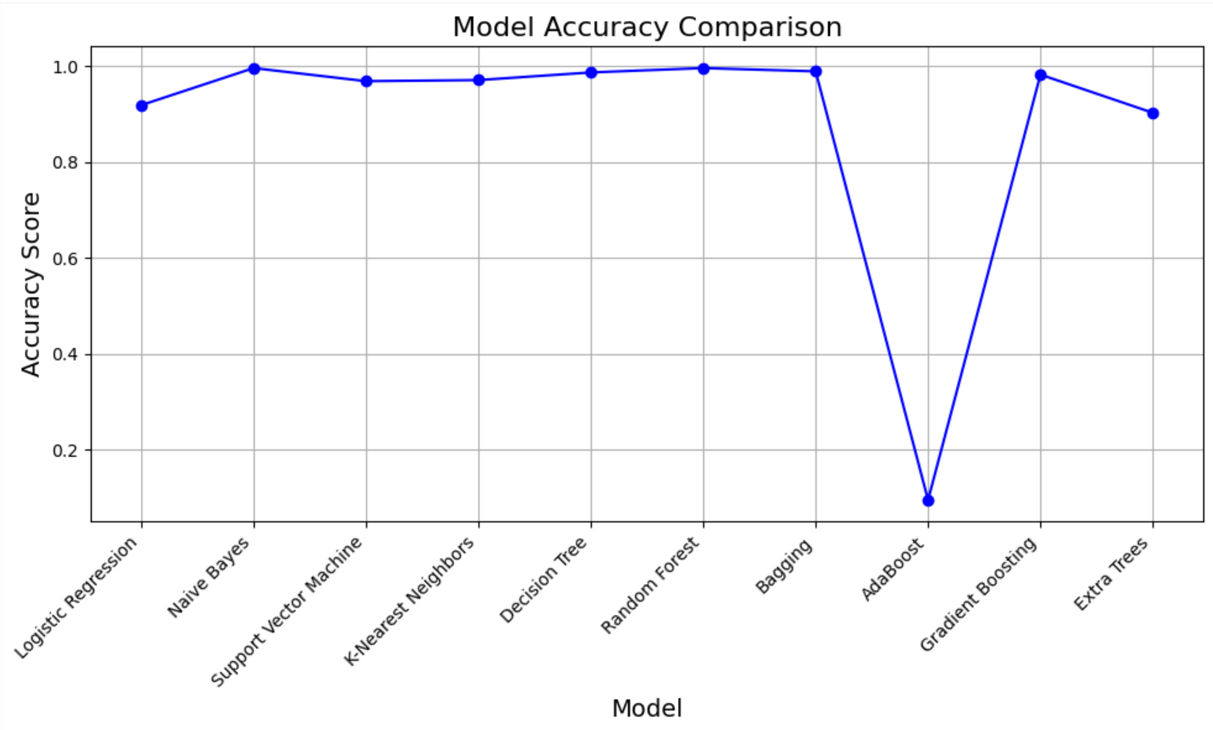

Fig: Model accuracy Comparison

Final Choice:
The Random Forest Classifier was selected for its superior accuracy, ability to handle non-linear relationships, and robust performance across diverse inputs.

4.1.3 Deploying the Model

The trained Random Forest model and scaler were saved as serialized files using Python's pickle library for integration into the Flask-based web application.

Files Created:

- model.pkl: Stores the trained Random Forest model.
- minmaxscaler.pkl: Contains the feature scaler used during data preprocessing.

4.2 Technical Approach

This section shows the technical methodology along with the corresponding code implementation for the Crop Recommendation System. The technical approach is further subdivided into subsections grouped around the project's pipeline.

Data Preprocessing

The data preprocessing stage involves preparing the dataset for machine learning by handling categorical values, scaling features, and splitting the data for training and testing.

Code:

```
# Encoding the target variable
crop_dict = {'rice': 1, 'maize': 2, ...}  # Crop mapping dictionary
crop['crop_num'] = crop['label'].map(crop_dict)

# Splitting data into features and target
X = crop.drop(['crop_num'], axis=1)
y = crop['crop_num']

# Splitting into training and test sets
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature Scaling using MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()
X_train = ms.fit_transform(X_train)
X_test = ms.transform(X_test)
```

What this code does:

- Encoding the crop labels into numeric values (crop_dict mapping).
- Applying MinMaxScaler for feature scaling.
- Splitting the dataset using train_test_split.

Model Training

The model training stage evaluates different machine learning algorithms and selects the best-performing one based on accuracy and robustness.

Code:

```
# Training different models
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)

# Testing the model
ypred = rfc.predict(X_test)
accuracy = accuracy_score(y_test, ypred)
print(f"Random Forest Accuracy: {accuracy}")

# Saving the model for deployment
import pickle
pickle.dump(rfc, open('model.pkl', 'wb'))
pickle.dump(ms, open('minmaxscaler.pkl', 'wb'))
```

What this code does:

- Defining and training models.
- Selecting the Random Forest Classifier based on evaluation metrics.

Prediction System

This stage involves building a predictive function that processes user inputs, scales them, and passes them through the trained model to generate recommendations.

Code:

```
# Predictive system function
def recommendation(N, P, K, temperature, humidity, ph, rainfall):
    features = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
    transformed_features = ms.transform(features)
    prediction = rfc.predict(transformed_features)
    return prediction[0]

# Example usage
N, P, K, temperature, humidity, ph, rainfall = 40, 50, 50, 30, 70, 6, 80
predicted_crop = recommendation(N, P, K, temperature, humidity, ph, rainfall)
print("{} is a best crop to be cultivated ".format(crop))
```

What this code does:

- The recommendation() function for crop prediction.
- Example usage with input values.

Summary of Code Usage

- **Data Preprocessing**: Use code from the .ipynb file for feature engineering and scaling.
- **Model Training**: Include training and serialization code for the Random Forest model.
- **Prediction System**: Adapt the recommendation() function for generating predictions.

Test Cases

Test ID	Scenario	Input Parameters	Expected Outcome	Status
T01	Valid Inputs	N=40, P=50, K=50, Temp=30, Hum=70, pH=6, Rain=80	Displays the best crop recommendation.	Passed
T02	Invalid Inputs (empty)	N=null, P=null, K=null, etc.	*Error*: "Please fill out this field"	Passed
T03	Outlier Inputs	N=200, P=300, K=500, Temp=-10, Hum=120, pH=14, rain = 1200	*Error*: "Inputs out of range"	Passed
T04	Edge Case	N=0, P=0, K=0, Temp=0, Hum=0, pH=0, Rain=0	*Recommended Crop*: No suitable crop	Passed

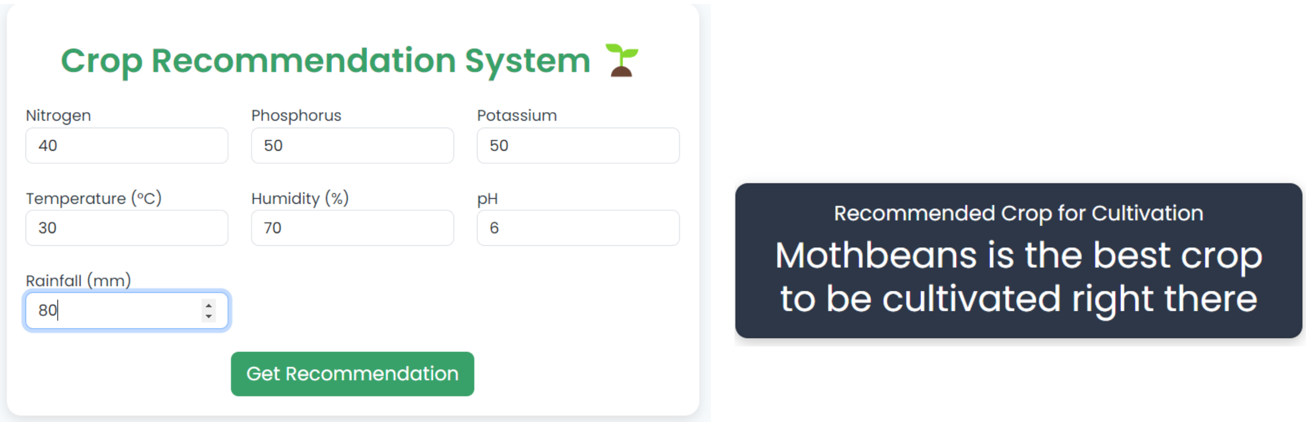

Fig: Test Case 01

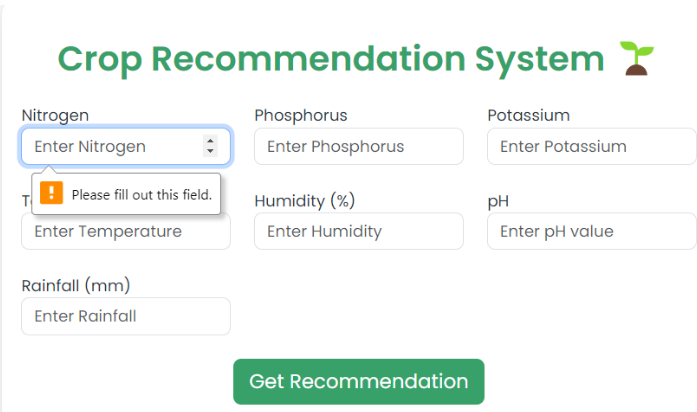

Fig: Test Case 02

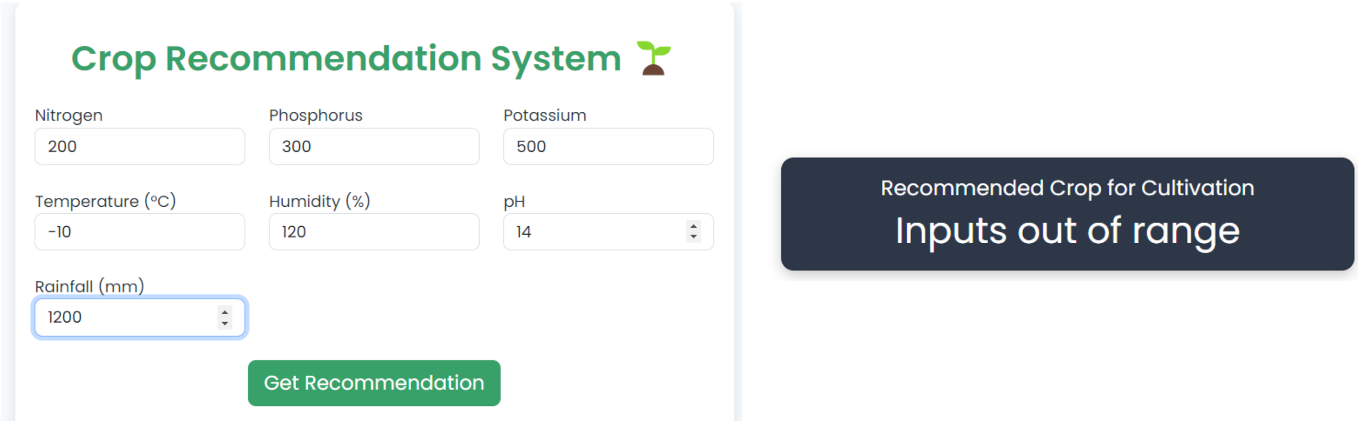

Fig: Test Case 03

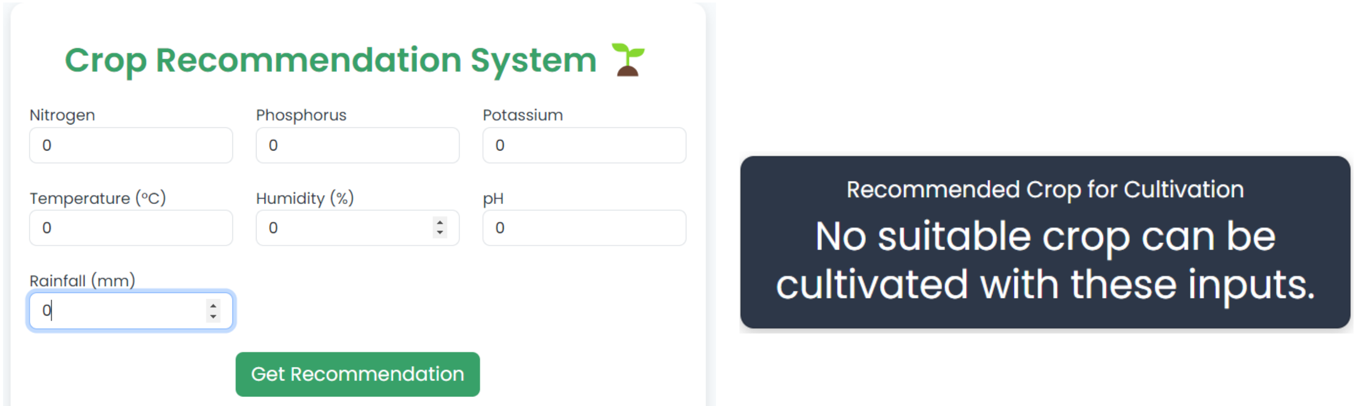

Fig: Test Case 04

4.3 Results and Screenshots

- **Results Achieved**

The system provided accurate crop recommendations based on user inputs.

- **Example:**
    - Inputs: Nitrogen = 40, Phosphorus = 50, Potassium = 50, Temperature = 40°C, Humidity = 20%, pH = 6.5, Rainfall = 100 mm.
    - Output: "KidneyBeans is the best crop to be cultivated right there."

Screenshots

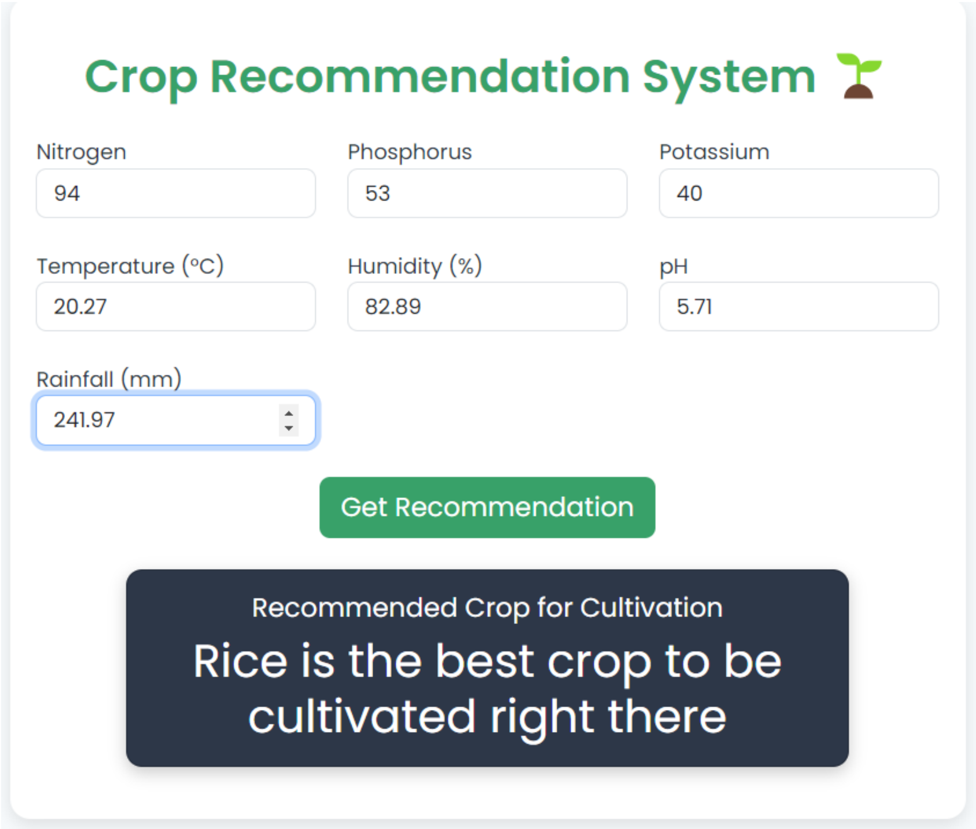

Fig: Example 1

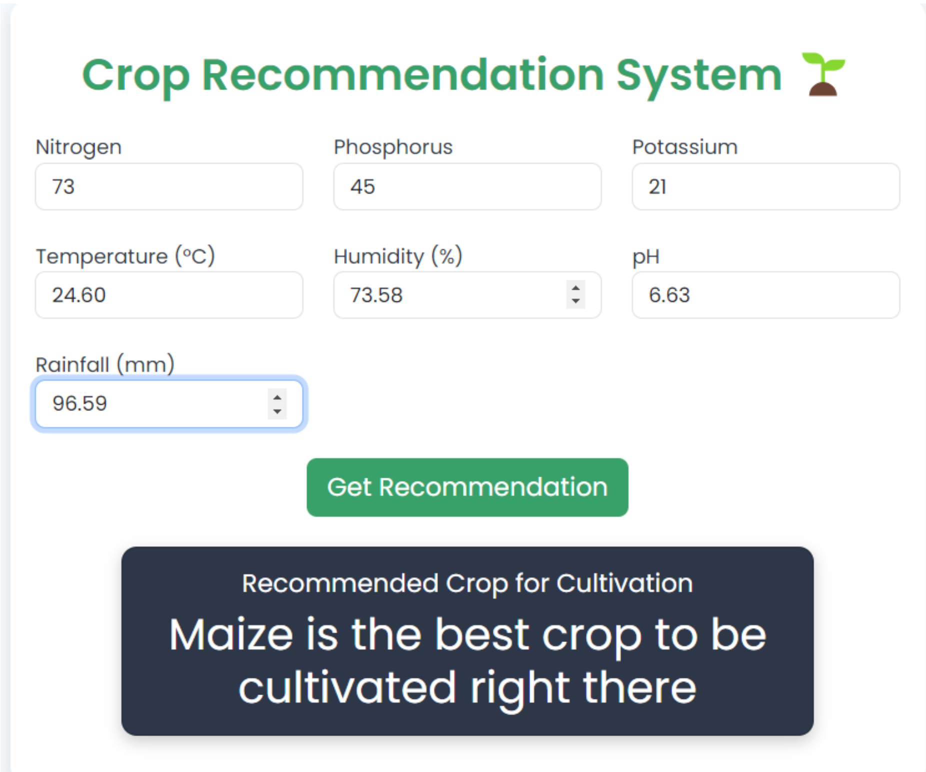

Fig: Example 2

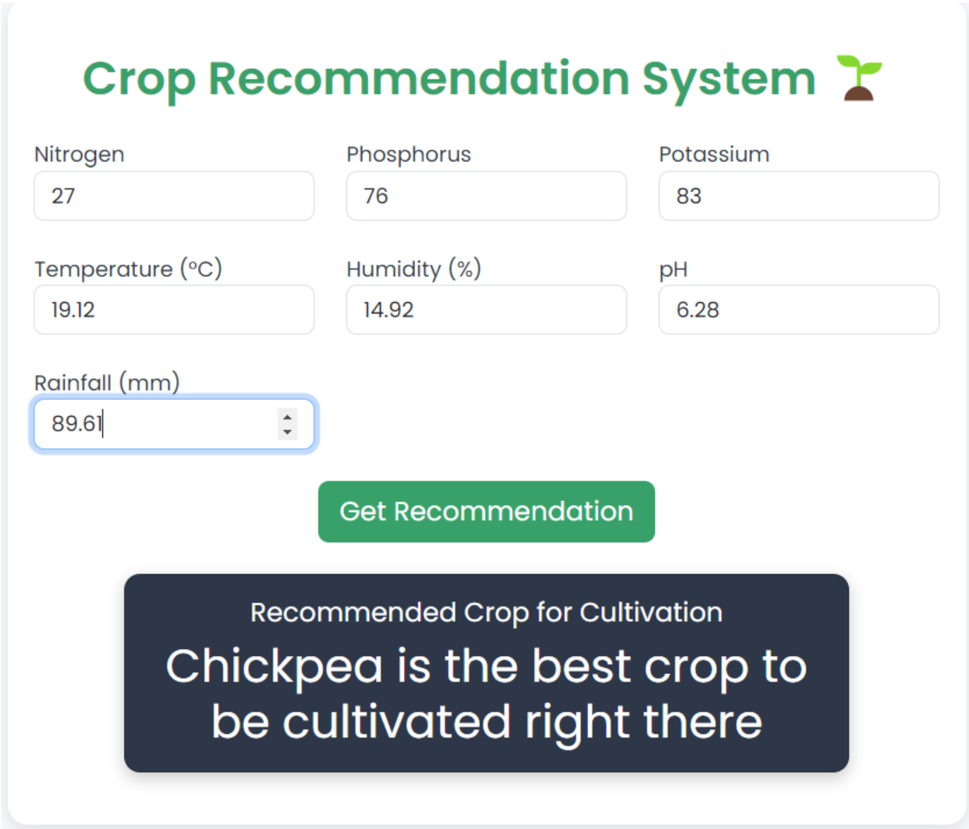

Fig: Example 3

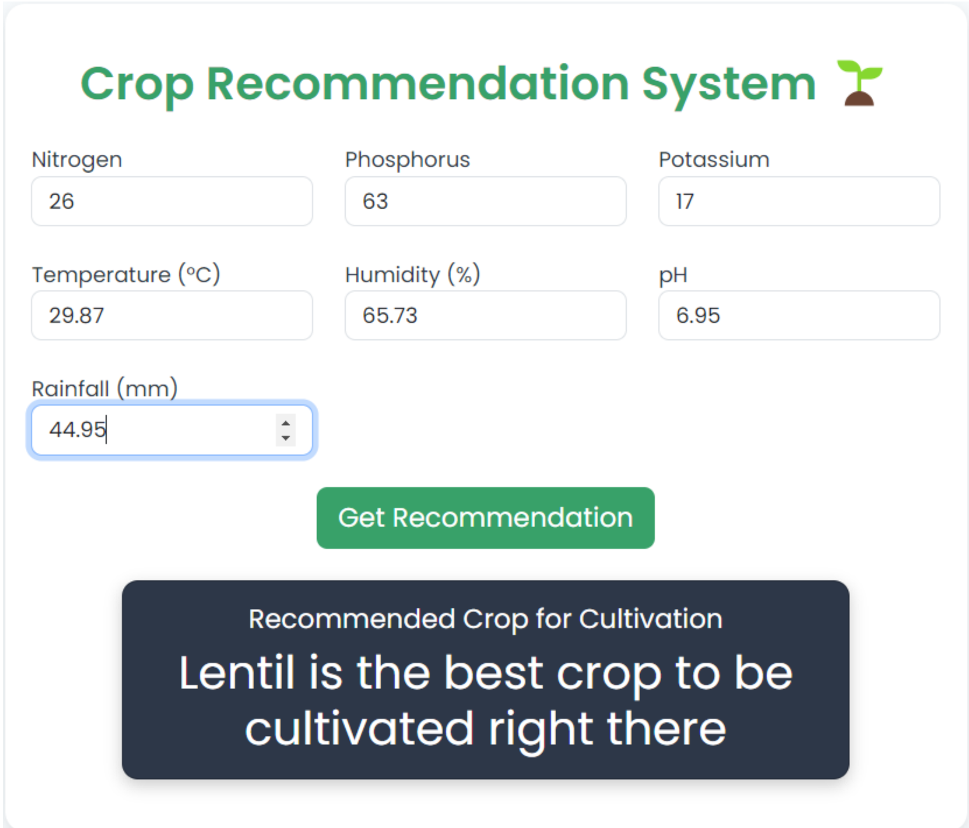

Fig: Example 4

4.4 Quality Assurance

4.4.1 Usability Testing

- The web application was tested with users of varying technical expertise, ensuring the interface was simple and intuitive.

4.4.2 Scalability Testing

- Simulated multiple user sessions to confirm the system could handle high traffic without performance degradation.

4.4.3 Standards Compliance

1. **Design Standards:** Followed best practices for responsive web design to ensure compatibility across devices.

2. **Coding Standards:** Wrote clean, modular code with meaningful variable names and proper documentation.

3. **Testing Standards:** Adhered to ISO and IEEE guidelines for system validation and testing.

The implementation of the Crop Recommendation System integrated advanced machine learning techniques with a robust web application framework. By focusing on rigorous testing and quality assurance, the system delivers accurate, reliable, and actionable recommendations to users, helping bridge the gap between technology and agriculture.

Chapter 5

Challenges and Problems

Building up this crop recommendation system was not easy and was packed with significant obstacles, and learning how to overcome them was the experience. Here's a briefing of some of the major obstacles we faced:

6.1 Data-Related Challenges

- **Imbalanced Dataset:** One of the first issues we encountered was the uneven distribution of data across different crops. This imbalance could lead to biased predictions, favoring certain crops over others. To address this, we had to carefully split and validate the data to maintain fairness in the model's outputs.

- **Handling Missing and Duplicate Data:** Like many real-world datasets, ours wasn't perfect. We found missing values and duplicate entries that could distort the results if left unaddressed. Cleaning the data by filling in the gaps and removing duplicates became an essential part of the preprocessing step.

6.2 Feature Engineering Challenges

- **Scaling the Features:** The data had variables with wildly different ranges—such as nitrogen levels, rainfall, and pH. If not scaled properly, this could skew the model's performance. Using tools like MinMaxScaler, we ensured all features were on the same scale.

- **Identifying Relevant Features:** Determining which features had the most impact on crop predictions was tricky. Factors like temperature and humidity were clearly important, but the challenge was to let the data guide us while also applying our domain knowledge.

6.3 Model Selection Challenges

- **Choosing the Best Model:** We tested several machine learning models, including Logistic Regression, Naive Bayes, SVM, Decision Trees, and Random Forest. While both Random Forest and Naive Bayes delivered high accuracy, Random Forest was ultimately chosen because it handled complex data relationships better.

- **Overfitting Issues:** During training, some models, especially Decision Trees and Gradient Boosting, overfit the data, performing well on training but poorly on testing. Tweaking the parameters and using regularization techniques helped minimize this problem.

6.4 Deployment and Integration Challenges

- **Integrating the Model with Flask:** Once the model was ready, embedding it into a Flask-based web application presented new challenges. Ensuring that data preprocessing in the app matched the pipeline used during training required careful attention.

- **Managing Files and Model Storage:** Saving the trained model and ensuring it could be seamlessly loaded in the Flask app was another hurdle. We used tools like pickle but had to troubleshoot compatibility issues during the initial stages.

6.5 Computational Challenges

- **Resource Limitations:** Training models like Random Forest and Gradient Boosting demanded significant computational power. This was especially challenging on limited hardware, so optimizing the data handling process became crucial.

- **Evaluation Time:** Comparing the performance of multiple models was time-consuming, particularly for complex algorithms. We had to carefully balance the need for thorough evaluation with time constraints.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

One excellent example would be a crop recommendation system for a farmer to determine what crop to grow by basing it on environmental and soil data using models such as logistic regression, naive Bayes, and random forest. Here we have chosen the latter because it performs best with complex relationships. Despite the above challenges such as data imbalance and overfitting, the system met all the objectives and provided a strong yet practical solution to agricultural optimization.

6.2 Future Scope

Another potential that can be maximized is when real-time data from weather APIs and IoT sensors are used to make the recommendation more dynamic. The accuracy of the forecasts also increases with the expansion of dataset capability for regional variations and granular details. Finally, a user interface that can be friendlier or perhaps even mobile or voice-enabled applications can improve the accessibility. In the future, these advances would include more sophisticated algorithms like deep learning and even include satellite images for even more accurate crop recommendations.

References

[1] Ramesh, S., & Sivakumar, S. (2019). Machine learning algorithms for crop prediction and recommendation in precision agriculture. *Agricultural Informatics Journal*, 14(1), 45-59.

[2] Patil, S. & Patil, P. (2020). A review of crop recommendation systems using machine learning techniques. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 45-50.

[3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

[4] Zhang, H. (2004). The optimality of naive Bayes. In *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference* (pp. 112-116). Menlo Park, CA: AAAI Press.

[5] Lin, C., & Lin, S. (2019). Support vector machine classification in agriculture: Applications and challenges. *International Journal of Agricultural Technology*, 15(6), 1315-1325.

[6] Gupta, A., & Gupta, S. (2021). Crop recommendation system based on environmental and climatic conditions using machine learning. *International Journal of Computer Applications*, 175(12), 11-17.

[7] Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming – A review. *Agricultural Systems*, 153, 69-80.

INDIVIDUAL CONTRIBUTION REPORT:

CROP RECOMMENDATION SYSTEM

SULAV BHANDARI
21053470

Abstract:
Crop Recommendation System is one that forecasts the type of crops and climatic conditions in a given location using machine learning techniques. This project was managed collaboratively, with team members working on different tasks like preprocessing data, selection of models, and preparing the report.

Individual Contribution and Findings:
I was responsible for data collection and cleaning, ensuring that the dataset was accurate and ready for model training. This involved handling missing values, normalizing data, and removing outliers. I also contributed to feature selection and model evaluation.

Individual Contribution to Project Report Preparation:
I contributed to the Introduction and Dataset Description sections, outlining the dataset used and explaining how it was prepared for analysis.

Individual Contribution for Project Presentation and Demonstration:
I contributed to explaining the data preparation process in the presentation, including how we handled missing data and selected features for model training.

Full Signature of Supervisor: …………………………….

Full signature of the student: ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

CROP RECOMMENDATION SYSTEM

RANJAN SRIVASTAV
21053402

Abstract:
Crop Recommendation System is one that forecasts the type of crops and climatic conditions in a given location using machine learning techniques. This project was managed collaboratively, with team members working on different tasks like preprocessing data, selection of models, and preparing the report.

Individual Contribution and Findings:
I was responsible for model selection and training. I tested multiple machine learning models like Logistic Regression and Naive Bayes and compared their performance using accuracy and confusion matrices. After evaluating several models, I selected Random Forest Classifier for its superior performance.

Individual Contribution to Project Report Preparation:
I contributed to the Model Selection and Training section, detailing the models we tested and their performance metrics. I also helped write the Model Evaluation portion of the report.

Individual Contribution for Project Presentation and Demonstration:
I presented the model evaluation results and explained the reasons for choosing the Random Forest Classifier. I also helped demonstrate the system during the presentation.

Full Signature of Supervisor: ……………………………

Full signature of the student: ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

CROP RECOMMENDATION SYSTEM

CHANDRA BAHADUR CHHETRI
21053415

Abstract:
Crop Recommendation System is one that forecasts the type of crops and climatic conditions in a given location using machine learning techniques. This project was managed collaboratively, with team members working on different tasks like preprocessing data, selection of models, and preparing the report.

Individual Contribution and Findings:
I focused on feature engineering and model optimization. I created new features from the dataset and applied techniques like grid search and cross-validation to improve model performance. My work ensured that the Random Forest model was optimized for the best possible accuracy.

Individual Contribution to Project Report Preparation:
I contributed to the Feature Engineering and Model Optimization sections of the report, explaining the features selected for training and the optimization techniques used.

Individual Contribution for Project Presentation and Demonstration:
I explained the feature engineering process in the presentation and demonstrated how optimized models performed better in predictions.

Full Signature of Supervisor: ……………………………

Full signature of the student: ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

CROP RECOMMENDATION SYSTEM

SAPHAL PANTHA
21053458

Abstract:
Crop Recommendation System is one that forecasts the type of crops and climatic conditions in a given location using machine learning techniques. This project was managed collaboratively, with team members working on different tasks like preprocessing data, selection of models, and preparing the report.

Individual Contribution and Findings:
I worked on the front-end development and system integration. I created a web-based user interface using HTML and CSS for the recommendation system. I integrated the machine learning model with the front-end to provide predictions in real-time.

Individual Contribution to Project Report Preparation:
I contributed to the System Architecture and User Interface sections of the report, detailing the design of the system and how the user interacts with the recommendation system.

Individual Contribution for Project Presentation and Demonstration:
I demonstrated the web interface during the presentation, showing how users can input environmental parameters and receive crop recommendations based on the trained model.

Full Signature of Supervisor: …………………………….

Full signature of the student: ……………………………..

TURNITIN PLAGIARISM REPORT

(This report is mandatory for all the projects and plagiarism must be below 25%)

"CROP RECOMMENDATION SYSTEM"

ORIGINALITY REPORT

7%	6%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to KIIT University Student Paper	2%
2	lujun995.github.io Internet Source	1%
3	www.worldleadershipacademy.live Internet Source	1%
4	huggingface.co Internet Source	1%
5	link.springer.com Internet Source	<1%
6	www.tnsroindia.org.in Internet Source	<1%
7	"Merged Papers", 2020 7th International Conference on Smart Structures and Systems (Icsss), 2020 Publication	<1%
8	www.ir.juit.ac.in:8080 Internet Source	<1%
9	www.ijraset.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		