CROP MASTER
An AI - Driven Smart Farming Bot

A MINI PROJECT REPORT

Submitted by

JAYANESH (221801020)
SHANMUGASHREE M (221801049)

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE
AND DATA SCIENCE
ANNA UNIVERSITY, CHENNAI – 602 105
NOV-2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this report titled "**CropMaster: An AI-Driven Smart Farming Bot**" is the bonafide work of **JAYANESH D (221801020) and SHANMUGASHREE M (221801049)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	SIGNATURE
Dr. J.M. Gnanasekar M.E., Ph.D.,	**Dr. P. Indira Priya M.E., Ph.D.,**
Professor and Head	Professor
Department of AI&DS	Department of AI&DS
Rajalakshmi Engineering College	Rajalakshmi Engineering College
Chennai – 602 105	Chennai – 602 105

Submitted for the project viva-voce examination held on _____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. J.M. GNANASEKAR., M.E., Ph.D.,** Head of the Department, Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor and coordinator, **Dr. P. INDIRA PRIYA M.E., Ph.D., Professor**, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for her valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

The agriculture sector is crucial to both the Indian economy and global food security, often lacks accessible, accurate information for farmers. **Crop Master: An AI-Driven Smart Farming Bot,** addresses this gap by providing small-scale farmers with a crop recommendation system designed to improve crop management and productivity. By analyzing soil properties through real-time soil testing and leveraging machine learning models like Random Forest, Crop Master enables precise crop recommendations that align with soil suitability. This AI-driven solution empowers farmers by doubling their yield potential, enhancing income, and fostering sustainable agricultural practices. The system offers a user-friendly interface, enabling farmers to access insights without extensive technical knowledge. Challenges and future directions in agricultural technology integration are explored to increase Crop Master's impact on farming efficiency and sustainability.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Agriculture is the foundation of India’s economy, supporting a vast portion of the population and ensuring food security. Small-scale farmers, who form the majority, often face challenges in accessing reliable information on crop management and soil health, limiting their productivity and income. Addressing these issues is essential for promoting sustainable practices that enhance yields and improve farmers' livelihoods.

With advancements in AI and real-time soil analysis, technology offers precise, data-driven solutions for agriculture. By leveraging machine learning models and soil testing, farmers can receive crop recommendations tailored to their unique soil properties, enabling efficient, informed decisions. These tools empower farmers to boost productivity and income while promoting environmentally responsible practices.

1.2 NEED FOR THE STUDY

Small-scale farmers often lack access to timely, accurate information that can significantly improve crop yield and income. Traditional farming practices, while familiar, frequently fall short in precision, leading to suboptimal productivity and inefficient resource management. Addressing these limitations is essential for advancing sustainable agriculture and improving farmers' livelihoods.

The need for this study stems from the potential of AI-driven crop recommendations and soil analysis to bridge farmers' information gaps. By providing tailored insights, farmers can make decisions aligned with their soil conditions, boosting efficiency and sustainability. This approach supports productivity while promoting environmentally responsible practices.

1.3 OVERVIEW OF THE PROJECT

Crop Master: An AI-Driven Smart Farming Bot, is designed to address the challenges faced by small-scale farmers in optimizing crop yield and income. By incorporating AI and machine learning techniques, the bot analyzes real-time soil data to recommend crops that are best suited to specific soil properties. This targeted approach enables farmers to make informed choices, resulting in better productivity and efficient resource utilization.

The system leverages the Random Forest algorithm to process soil data and generate precise crop recommendations. Through user-friendly interfaces, farmers can access insights without the need for extensive technical knowledge, making the tool accessible to a wider audience. The bot's recommendations are based on data patterns derived from extensive soil and crop analysis, ensuring that suggestions are accurate and impactful.

In addition to increasing yield, the project promotes sustainable agricultural practices by optimizing resource use. Crop Master empowers small-scale farmers with actionable data, contributing to improved income and environmental sustainability.

1.4 OBJECTIVES OF THE STUDY

The objectives of this project are:

- Increase crop yields with precise, AI-driven recommendations based on real-time soil data.
- Improve resource management by optimizing water, fertilizer, and pesticide use.
- Support sustainable practices that maintain soil health and reduce environmental impact.
- Enable farmers to make timely and informed decisions for better outcomes.
- Enhance profitability for small-scale farmers by improving crop quality and efficiency.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

A literature survey on AI-driven crop recommendation systems reveals the growing role of technology in optimizing agricultural practices. Studies emphasize the importance of data-driven insights, particularly in small-scale farming, where traditional methods often limit productivity and sustainability. Research highlights machine learning and soil analysis as key tools for improving crop yield through targeted recommendations.

Additionally, advancements in soil testing and machine learning algorithms have paved the way for precise crop suggestions based on region-specific data. These innovations allow farmers to adopt efficient, sustainable practices tailored to local soil conditions, ultimately enhancing crop output and resource management in agriculture.

2.2 FRAMEWORK OF LCA

The framework of Life Cycle Assessment is essential for evaluating the environmental impacts associated with all stages of a product's life, from raw material extraction to disposal. In agriculture, LCA helps assess the ecological footprint of farming practices, including crop production, water usage, and waste management. By examining each stage, LCA identifies areas for improvement, promoting sustainable practices and resource efficiency.

1. Goal and Scope Definition

In this phase, the objectives of the LCA are established, defining the study's boundaries and functional units. For agriculture, the goal may be to evaluate crop production impacts or optimize resource usage. Setting clear objectives allows for a focused analysis, ensuring that the results align with sustainable development goals.

2. Inventory Analysis

This step involves collecting data on all inputs and outputs involved in farming, such as energy, water, fertilizers, and emissions. By quantifying these elements, inventory analysis provides a detailed picture of resource consumption and waste generation. In agriculture, it helps pinpoint resource-intensive areas that could benefit from improved management practices.

3. Impact Assessment

Impact assessment translates inventory data into environmental consequences, such as greenhouse gas emissions, water scarcity, and soil degradation. This analysis enables an understanding of how farming practices affect the environment. In agriculture, the impact assessment can guide choices that reduce negative outcomes, supporting ecosystem health.

4. Interpretation

In the final phase, the LCA findings are analyzed to identify potential improvements in farming practices. This interpretation helps inform decisions aimed at reducing environmental impacts and optimizing sustainability. For agriculture, it suggests actionable insights, like adopting low-impact fertilizers or water-saving techniques, to achieve a more sustainable farming approach.

5. Life Cycle Impact Mitigation

Mitigation focuses on identifying alternative farming techniques or inputs that can lower the environmental impact identified in the impact assessment. In agriculture, this may involve switching to renewable energy sources, reducing chemical pesticide use, or integrating practices like agroforestry. These changes help in lowering emissions, conserving water, and protecting soil and biodiversity.

6. Sensitivity and Uncertainty Analysis

This subtopic addresses the variability and uncertainty within the LCA data and results, often due to factors like changing weather patterns or varying soil conditions. Sensitivity analysis in agriculture allows for testing different scenarios, such as varying water or fertilizer use, to understand how changes affect overall environmental impacts. By accounting for uncertainties, it ensures that the recommendations are robust and adaptable to varying conditions.

7. Comparative Analysis

Comparative analysis assesses different farming practices or crops by applying the LCA framework to each. This comparison helps determine which approach or crop type has the lowest environmental impact, guiding farmers in selecting more sustainable options. It may compare traditional vs. organic farming or evaluate various crop rotations to find the least resource-intensive option.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

Crop Recommendation commonly depend on broad regional data, providing only generalized crop suggestions. These systems often lack the precision to consider specific soil properties, nutrient levels, and climate variations unique to each farm plot, which are critical for optimizing crop yields. Traditional methods primarily involve manual soil testing and expert consultations, making the process time-intensive and costly, especially for small-scale farmers with limited access to such resources.

While some modern systems have started integrating machine learning techniques, such as decision trees, support vector machines, and random forest algorithms, they typically rely on static historical data rather than real-time inputs. As a result, the crop recommendations may not reflect recent changes in soil health, seasonal variations, or immediate environmental factors that could impact crop suitability. These limitations make it difficult for farmers to achieve precise, adaptive recommendations, especially in dynamic conditions.

Moreover, the lack of real-time monitoring and automated recommendations in most existing systems restricts farmers' ability to respond quickly to changing conditions or pest threats. In regions with diverse soil types and microclimates, these traditional systems fall short in delivering tailored insights that can help maximize productivity and sustainability.

Consequently, there is a growing need for AI-driven, real-time solutions that can offer accurate, site-specific recommendations, empowering farmers with data-driven, adaptive tools for effective crop management.

3.2 PROPOSED SYSTEM

The proposed system aims to overcome the limitations of existing crop recommendation models by incorporating AI-driven insights and real-time soil analysis, providing small-scale farmers with accurate, data-backed recommendations. By focusing on real-time data collection and advanced analytics, this system empowers farmers with tailored guidance that adapts to their specific soil conditions, climate factors, and crop needs, ultimately enhancing productivity and sustainability.

- **Soil and Crop Analysis:** Using historical and regional soil data, the system analyzes parameters such as soil pH, nutrient content, and texture to suggest crops best suited for specific soil profiles. By drawing on comprehensive datasets, the system can offer reliable recommendations that account for variations in soil properties across different regions, enhancing compatibility between soil conditions and crop requirements.
- **Model Training and Selection:** The system will train nine different machine learning models to identify the most suitable crop based on soil, weather, and environmental data. Models such as Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naïve Bayes, Gradient Boosting, Neural Networks, and ensemble methods will be evaluated. After comparing the performance metrics (accuracy, precision, recall) of each model, the best-performing model will be selected for deployment. This approach ensures high reliability and accuracy in crop recommendations.
- **User-Friendly Interface:** Designed with an easy-to-use interface, the system provides clear, actionable insights accessible through a mobile or web application. Farmers can view detailed crop recommendations, receive alerts for soil health maintenance, and access visual reports on soil trends. This accessibility is crucial for small-scale farmers who may not have advanced technical expertise, making the system more inclusive and practical for real-world agricultural use.

- **Sustainability and Environmental Focus:** The system promotes sustainable practices by recommending crops that require fewer resources and align with the natural soil profile. By reducing the need for excessive fertilizers and pesticides, the recommendations not only improve crop yield but also minimize environmental impact. The system's data-driven approach encourages eco-friendly farming, preserving soil health and supporting biodiversity over the long term.

- **Continuous Feedback and Adaptation:** The system integrates a feedback mechanism that learns from farmer inputs and past crop performance to refine its recommendations over time. As farmers implement the suggested crops and provide data on outcomes, the system adjusts its predictions, creating a self-improving cycle. This continuous feedback loop ensures the system remains accurate and increasingly effective, further optimizing agricultural productivity for small-scale farmers.

3.3 FEASIBILITY STUDY

The feasibility study for Crop Master assesses the project's practicality in terms of technical, economic, and operational factors, ensuring it is both viable and impactful for small-scale farmers. By examining these areas, we can better understand the system's potential to improve agricultural productivity sustainably.

Technical Feasibility

The project leverages existing soil and crop datasets along with machine learning algorithms to analyze patterns and recommend suitable crops. Since it doesn't require real-time soil analysis, the system remains accessible and cost-effective for small-scale farmers. The technical foundation includes readily available data-processing frameworks, ensuring the system's development and deployment are straightforward without extensive technological resources.

Economic Feasibility

One of the project's goals is to make crop recommendations accessible to small-scale farmers with limited financial resources. By using a dataset-based model, the system minimizes hardware costs associated with real-time monitoring and allows users to gain insights through a low-cost or subscription-based app. This approach provides economic value to farmers by enhancing productivity without substantial upfront investment, making the system a financially sustainable solution.

Operational Feasibility

The proposed system is designed to be user-friendly and requires minimal technical knowledge, making it highly feasible for farmers with varying degrees of digital literacy. By providing crop recommendations based on regional datasets, it aligns well with farmers' needs and existing practices, promoting seamless integration into day-to-day operations. Additionally, the platform's modular design allows for easy updates and customization, ensuring that the system remains relevant and operationally efficient.

Legal and Environmental Feasibility

The project considers data privacy standards and regulatory guidelines, ensuring compliance with legal requirements surrounding agricultural data use. From an environmental perspective, the system promotes sustainable farming practices by recommending crops suited to local soil and climate conditions. By encouraging responsible resource management and reducing chemical inputs, the system contributes positively to environmental conservation and sustainable agriculture.

CHAPTER 4
SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Processor : i3 & Above
- RAM : 4GB & Above
- Storage : MongoDB

Network Requirements

- High-speed internet connection for accessing cloud services (if needed) and data transfer.
- Secure local network for on-premises

4.2 SOFTWARE REQUIREMENTS

- Operating System : Windows XP/ 7 & Above, Linux and Mac
- Coding Language : Python
- Frontend : HTML, CSS, Javascript, Bootstrap,
- Backend : Flask
- Database : MongoDB
- ML Libraries : scikit-learn for building predictive models.
- Data Processing Libraries: Pandas, NumPy, SciPy
- Visualization Tools : Matplotlib, Seaborn

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

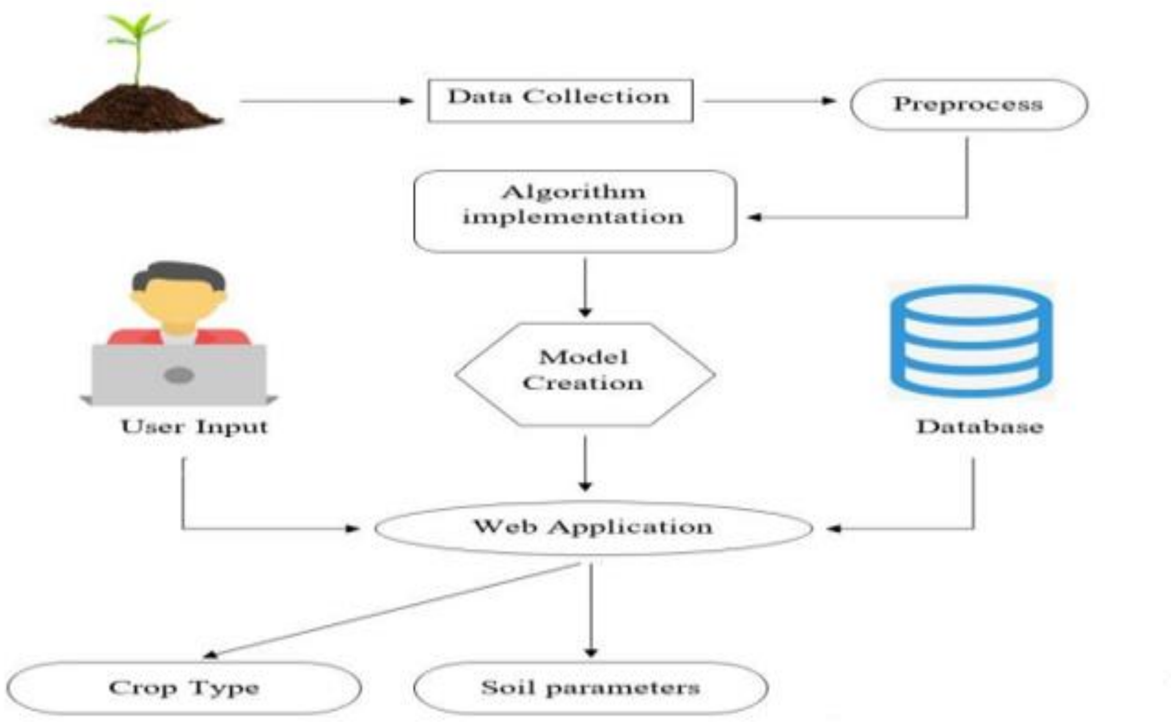

Figure 1: System Architecture

This Architectural diagram illustrates the workflow of a Crop Recommendation System. The system collects and processes agricultural data to provide farmers with suitable crop recommendations through a web application interface. It involves several stages, from data gathering to user interaction and output generation.

1. **Data Collection:** Begins with gathering relevant data, such as soil and environmental information.
2. **Preprocess:** The data is preprocessed to clean and prepare it for analysis.
3. **Algorithm Implementation:** Various algorithms are applied to create a predictive model.

4. **Model Creation:** The model is developed using the processed data, aiming to make accurate recommendations.
5. **Database:** Stores data and user inputs to support the model's training and user interactions.
6. **User Input:** Users, such as farmers, input relevant data into the system through a web application.
7. **Web Application:** Provides an interface for users to interact with the model and access recommendations.
8. **Outputs:** The system delivers recommendations on suitable crop types and soil parameters, helping users make informed decisions.

5.2 MODULE DESCRIPTION

This section provides an overview of each module involved in the crop prediction system, explaining how each module contributes to the system's functionality and how they interact with each other. Each module is structured to align with the system's design and enhance the predictive accuracy of crop recommendations.

5.2.1 Dataset Collection and Preprocessing

The crop prediction project dataset is sourced from Kaggle. After gathering, the dataset undergoes preprocessing to ensure data quality. Missing values are removed, and the remaining data is refined for accurate model training. This cleaned dataset serves as the foundation for reliable predictions.

5.2.2 Algorithm Selection and Implementation

Various machine learning algorithms were chosen to assess their effectiveness in predicting crop suitability. Each algorithm has unique characteristics suited to different types of data and distributions:

1. **Logistic Regression**: A linear model often used for classification. The formula:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}}$$

Here, β represents the model coefficients, optimizing probabilities for binary or multiclass classification.

2. **Decision Tree**: Constructs a tree structure where nodes represent features and branches represent decisions. It recursively splits data at points to maximize information gain or minimize Gini impurity.
3. **Random Forest**: An ensemble of decision trees. The formula for the decision function:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

where T is the number of trees, and ht(x)h_t(x)ht(x) is the output of each tree. Averaging multiple trees improves stability and accuracy.

4. **Support Vector Machine (SVM)**: A classifier that finds the optimal hyperplane to separate classes. The decision function:

$$f(x) = \text{sign}(w \cdot x + b)$$

aims to maximize the margin between classes.

5. **K-Nearest Neighbors (KNN)**: Classifies based on the majority label among the k-nearest samples. The formula for distance calculation (e.g., Euclidean distance) is:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

6. using Bayes' theorem. The formula for calculating posterior probability:

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

7. **Linear Discriminant Analysis (LDA)**: Projects data onto lower dimensions where separation between classes is maximized, using means and variances of each class.
8. **Bagging & Gradient Boosting Classifiers**: Ensemble methods that combine weak learners to improve overall model accuracy, reducing overfitting and improving generalization.

5.2.3 Model Training

In the model training phase, each algorithm was trained on preprocessed data to develop an effective crop recommendation system. Robustness was assessed using 4-fold cross-validation, a technique that divides the data into four subsets to test model performance across multiple data splits, helping avoid overfitting. The models were then fitted to the training data, adjusting parameters based on default or customized hyperparameters, to optimize performance. To facilitate easy reloading of trained models for future predictions, each was serialized and saved using pickle.

5.2.4 Model Evaluation

For evaluation, models were tested on the reserved test data, with performance measured through the accuracy score, which represents the proportion of correct predictions. Detailed metrics, including precision, recall, and F1 scores, were provided in classification reports to give a comprehensive view of each model's accuracy across classes. A visual comparison of model test accuracies was conducted using Seaborn's bar plot, highlighting the top-performing model based on test accuracy. This model, achieving the highest accuracy, was designated as the best model and may serve as the default recommendation model for future crop predictions.

5.2.5 Prediction and Recommendation

In this module, the preprocessed data is used to train the Gaussian Naive Bayes model, which achieved the highest accuracy among all evaluated models. This algorithm was chosen for its high performance in classifying crops based on environmental and soil conditions. The trained Gaussian Naive Bayes model operates using Bayes' Theorem, where it calculates the likelihood of each crop label given the user's input data and assigns a label based on posterior probabilities.

Upon user input, the data is scaled and passed through the model, predicting the crop type best suited to the conditions provided. This prediction is displayed as output on a web application built with Flask, enabling an interactive and accessible experience for users. The Flask framework serves as a user-friendly interface, allowing farmers and agricultural consultants to access real-time recommendations directly through the application. This structure makes the crop recommendation system not only accurate but also practical and accessible for everyday use.

CHAPTER 6

RESULTS AND DISCUSSIONS

The crop recommendation system effectively utilized multiple algorithms to identify the best model for recommending crops based on environmental and soil factors. After evaluating models like Logistic Regression, Decision Tree, Random Forest, SVM, KNN, Naive Bayes, LDA, Bagging, and Gradient Boosting, the Random Forest model emerged as the top performer, achieving the highest accuracy score. Cross-validation confirmed its robustness, indicating consistent performance across different data splits. The accuracy and classification report metrics showed that Guassian Naïve Bayes accurately classified crop types, with high precision and recall across classes. Visualizing test accuracies through a bar plot highlighted Guassian Naïve Bayes as the preferred model, making it suitable for practical crop prediction. Overall, the system provides an accessible, accurate tool for crop recommendation, supporting data-driven decisions for improved agricultural outcomes.

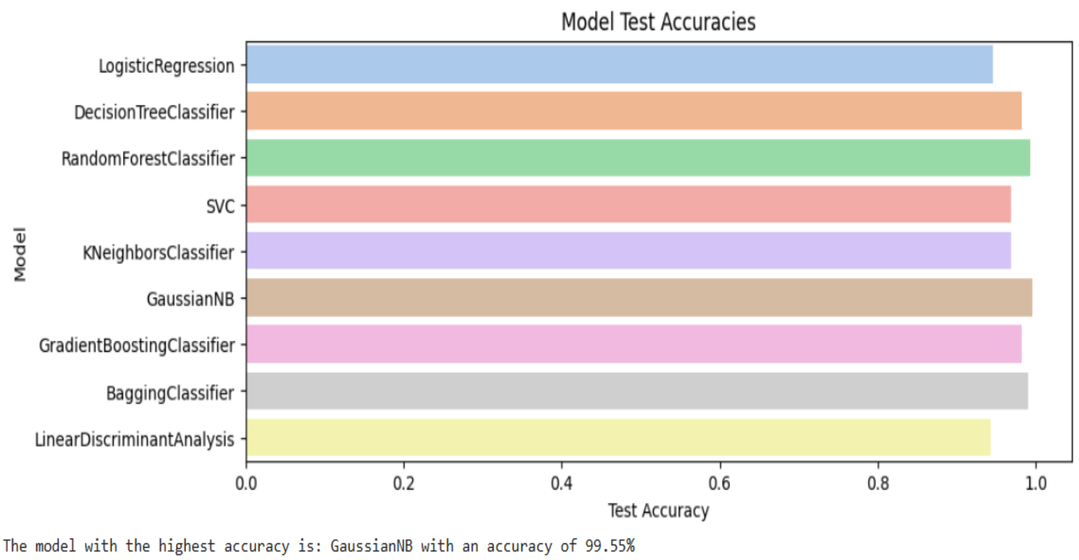

Figure 2: Model Performance

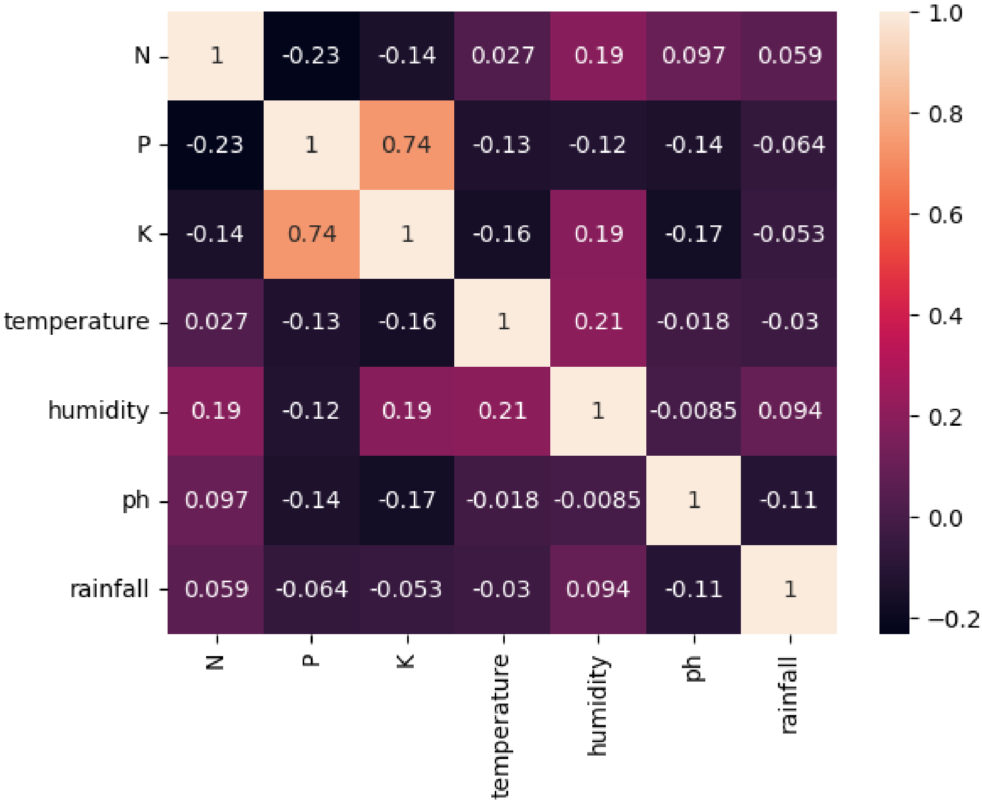

Figure 3: Correlation Heatmap For the distribution of dataset

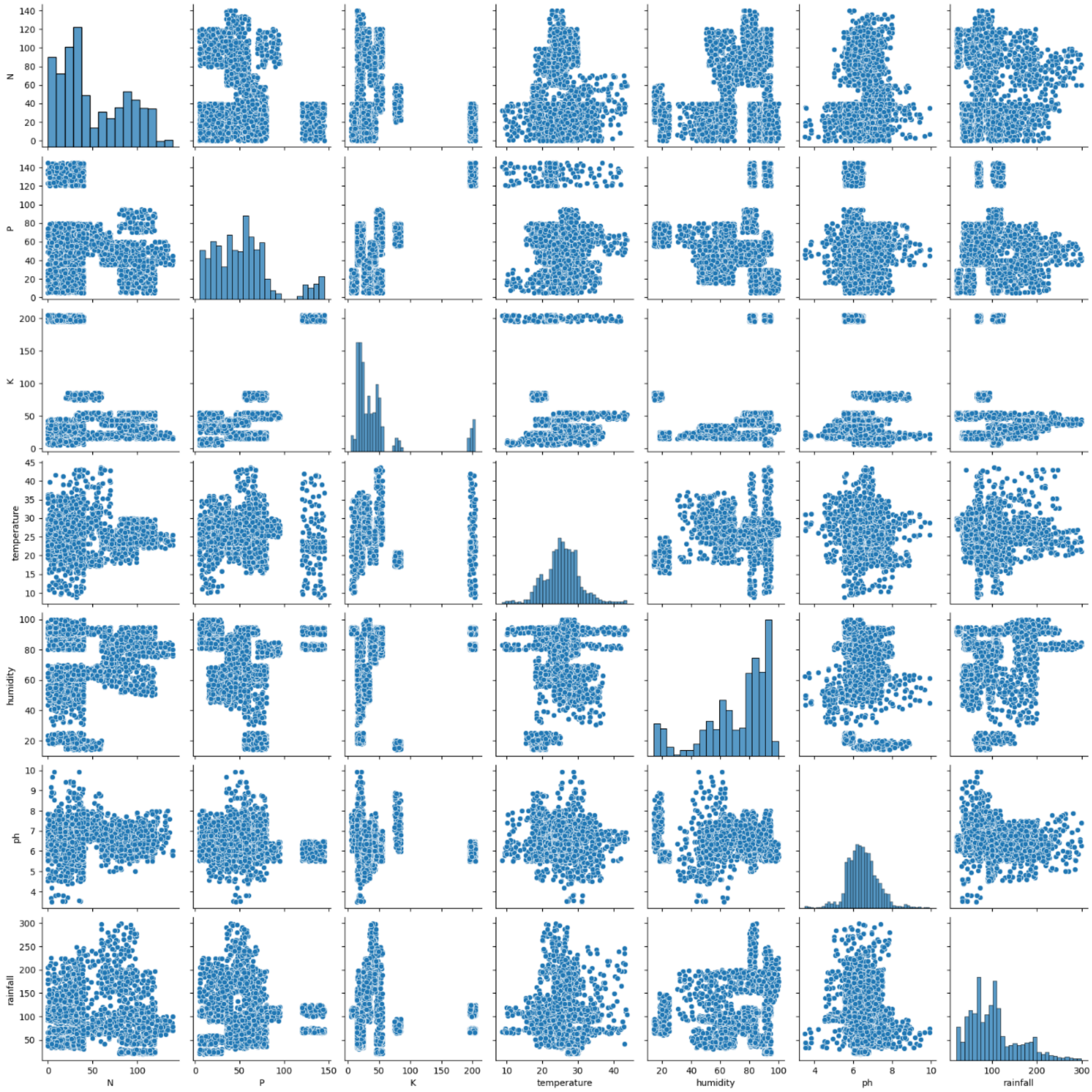

Figure 4: Pairplot of the dataset

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In conclusion, this crop prediction project effectively demonstrates the potential of machine learning to support agricultural decision-making. By leveraging the Random Forest algorithm, the system provides accurate crop recommendations based on soil and environmental parameters, bridging the gap between complex data analytics and practical agricultural applications. The project also highlights the importance of data preprocessing, algorithm selection, and user-friendly interfaces in creating an impactful solution for farmers and agricultural consultants. The integration of these elements into a web application allows for easy accessibility and real-time crop prediction, making advanced technology available to those who need it most.

7.2 FUTURE ENHANCEMENT

Future enhancements for the crop prediction system include incorporating real-time data from weather updates and soil sensors for dynamic, accurate recommendations. Expanding to additional algorithms like SVM and Neural Networks could further optimize performance for diverse conditions. Developing a mobile app with offline access would improve accessibility for farmers, especially in remote areas. Adding multilingual support would make the system more inclusive for non-English-speaking users. Finally, implementing a feedback loop would enable continuous learning, allowing the model to adapt to local conditions for more relevant recommendations.

7.3 APPENDIX

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split, cross_val_score
import warnings
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.metrics import accuracy_score, classification_report
import pickle

crop_recommendation_data = pd.read_csv("crop_recommendation.csv")
crop_recommendation_data.head()
crop_recommendation_data.tail()
crop_recommendation_data.describe()
crop_recommendation_data.size
crop_recommendation_data.shape
crop_recommendation_data.info()
crop_recommendation_data['label'].unique()
crop_recommendation_data['label'].value_counts()
```

1. DATA PREPROCESSING MODULE

```
missing_values = crop_recommendation_data.isnull().sum()
missing_values
features = crop_recommendation_data.drop(['label'], axis = 1)
target = crop_recommendation_data['label']
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)
with open('minmax_scaler.pkl', 'wb') as file:
    pickle.dump(scaler,file)
features_dataframe_scaled = pd.DataFrame(features_scaled,
columns=features.columns)
X_train, X_test, y_train, y_test = train_test_split(features_dataframe_scaled,
target, test_size=0.2, random_state=42)
print("Training Set - X:", X_train.shape)
print("Training Set - y:", y_train.shape)
print("Testing Set - X:", X_test.shape)
print("Testing Set - y:", y_test.shape)
```

2. MODEL BUILDING AND EVALUATION MODULE

```
accuracy_over_testdata = []
model_names = []
def crop_recommendation(algorithm, X_train, X_test, y_train, y_test):
    model_name = type(algorithm).__name__
    model_names.append(model_name)
    algorithm.fit(X_train, y_train)
    predictions = algorithm.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)
```

```
    print("Algorithm:", model_name)
    print("\n Accuracy Score Over Test Data:", accuracy)
    print("\n Classification Report for Test Data:\n",classification_report(y_test,
predictions) )
    accuracy_over_testdata.append(accuracy)
    print("\n Cross Validation Score:\n", cross_val_score(algorithm,
features_dataframe_scaled, target, cv=4, scoring = "accuracy"))
```

1. Logistic Regression

```
logistic_model = LogisticRegression(penalty='l1', solver='liblinear', random_state=42)
crop_recommendation(logistic_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationlog2.pkl', 'wb') as file:
    pickle.dump(logistic_model, file)
```

2. Decision Tree

```
decision_tree_model = DecisionTreeClassifier(min_samples_split=10,
random_state=42)
crop_recommendation(decision_tree_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationdecision.pkl', 'wb') as file:
    pickle.dump(decision_tree_model, file)
```

3. Random Forest

```
random_forest_model = RandomForestClassifier(
    random_state=44,    n_estimators=10,    max_depth=10,
    min_samples_split=5,    min_samples_leaf=2,    max_features='sqrt'
crop_recommendation(random_forest_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationrandom.pkl', 'wb') as file:
    pickle.dump(random_forest_model, file)
```

4. Support Vector Machine (SVM)

```
svm_model = SVC(C=1.0, kernel='rbf', random_state=42)
crop_recommendation(svm_model, X_train, X_test, y_train, y_test)
```

```
with open('crop_recommendationsvm.pkl', 'wb') as file:
    pickle.dump(svm_model, file)
```

5. K-Nearest Neighbors (KNN)

```
knn_model = KNeighborsClassifier(n_neighbors=5)
crop_recommendation(knn_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationknn.pkl', 'wb') as file:
    pickle.dump(knn_model, file)
```

6. Naive Bayes

```
nb_model = GaussianNB()
crop_recommendation(nb_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationnb.pkl', 'wb') as file:
    pickle.dump(nb_model, file)
```

7. Linear Discriminant Analysis (LDA)

```
lda_model = LinearDiscriminantAnalysis()
crop_recommendation(lda_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationlda.pkl', 'wb') as file:
    pickle.dump(lda_model, file)
```

8. Bagging Classifier

```
 bagging_model = BaggingClassifier(random_state=42)
crop_recommendation(bagging_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationbagging.pkl', 'wb') as file:
    pickle.dump(bagging_model, file)
```

9. **Gradient Boosting Classifier**

```
gb_model = GradientBoostingClassifier(random_state=42)
crop_recommendation(gb_model, X_train, X_test, y_train, y_test)
with open('crop_recommendationgb.pkl', 'wb') as file:
    pickle.dump(gb_model, file)
```

3. MODEL PERFORMANCE

```
print(f"Length of model_names: {(model_names)}")
print(f"Length of accuracy_over_testdata: {(accuracy_over_testdata)}")
df = pd.DataFrame({'Model': model_names, 'Test Accuracy':
accuracy_over_testdata})
colors = sns.color_palette("pastel", len(model_names))
plt.figure(figsize=(10, 4))
ax = sns.barplot(x='Test Accuracy', y='Model', data=df, palette=colors)
ax.set_title('Model Test Accuracies')
ax.set_xlabel('Test Accuracy')
plt.show()
max_accuracy = max(accuracy_over_testdata)
best_model = model_names[accuracy_over_testdata.index(max_accuracy)]
print(f"The model with the highest accuracy is: {best_model} with an
accuracy of {max_accuracy * 100:.2f}%")
```

4. MODEL PREDICTION

```
with open('crop_recommendationnb.pkl', 'rb') as file:
    model_loaded = pickle.load(file)
with open('minmax_scaler.pkl', 'rb') as file:
    scaler = pickle.load(file)
new_data = [[71, 41, 43, 22.3, 80, 6, 200]]
scaled_data = scaler.transform(new_data)
prediction = model_loaded.predict(scaled_data)
print("Predicted Crop:", prediction)
```

5. DATA VISUALIZATION

data_without_label = crop_recommendation_data.drop(["label"], axis = 1)

sns.pairplot(data_without_label)

plt.show()

sns.heatmap(data_without_label.corr(), annot=True)

7.4 OUTPUT SCREENSHOTS

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Figure 5: Head of the dataset

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

Figure 6: Tail of the dataset

	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

Figure 7: Description of the dataset

label	count
rice	100
maize	100
jute	100
cotton	100
coconut	100
papaya	100
orange	100
apple	100
muskmelon	100
watermelon	100
grapes	100
mango	100
banana	100
pomegranate	100
lentil	100
blackgram	100
mungbean	100
mothbeans	100
pigeonpeas	100
kidneybeans	100
chickpea	100
coffee	100

dtype: int64

Figure 8: Label of the dataset

	0
N	0
P	0
K	0
temperature	0
humidity	0
ph	0
rainfall	0
label	0

Figure 9: Output of Preprocessing

```
Training Set - X: (1760, 7)
Training Set - y: (1760,)
Testing Set - X: (440, 7)
Testing Set - y: (440,)
```

Figure 10: Splitting of the dataset

Algorithm	Accuracy Score (%)
Logistic Regression	94.55
Decision Tree Classifier	98.18
Random Forest Classifier	99.32
Support Vector Classifier (SVC)	96.82
K-Nearest Neighbors Classifier	96.82
Gaussian Naive Bayes Classifier	99.55
Gradient Boosting Classifier	98.18
Bagging Classifier	99.09
Linear Discriminant Analysis (LDA)	94.32

Figure 11: Model Accuracy Score

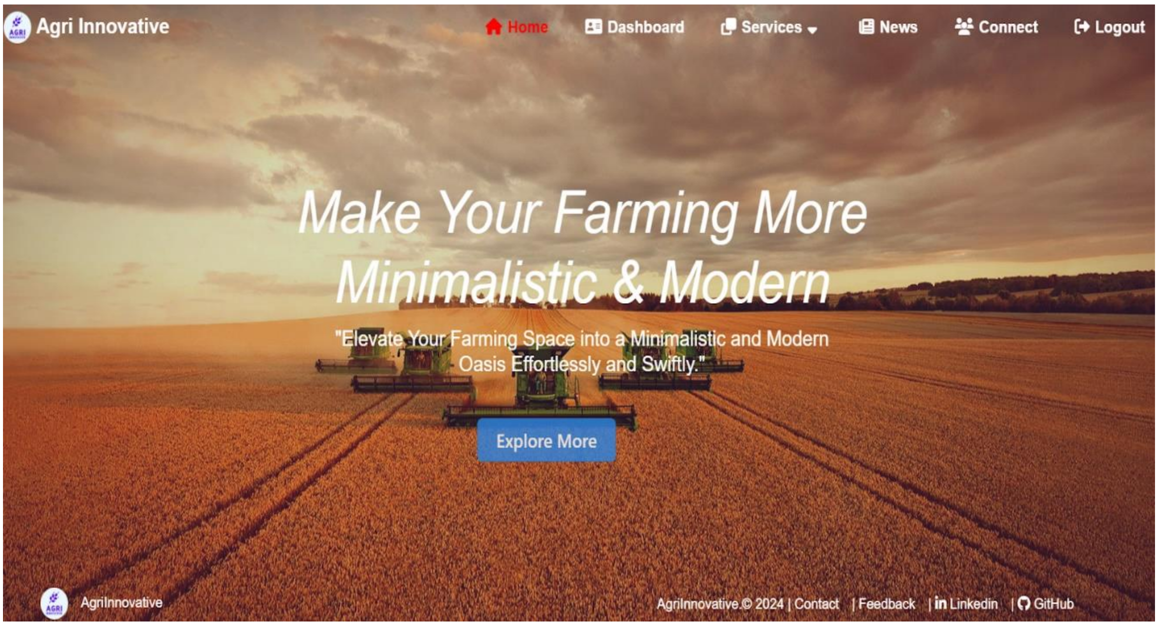

Figure 12: Home page

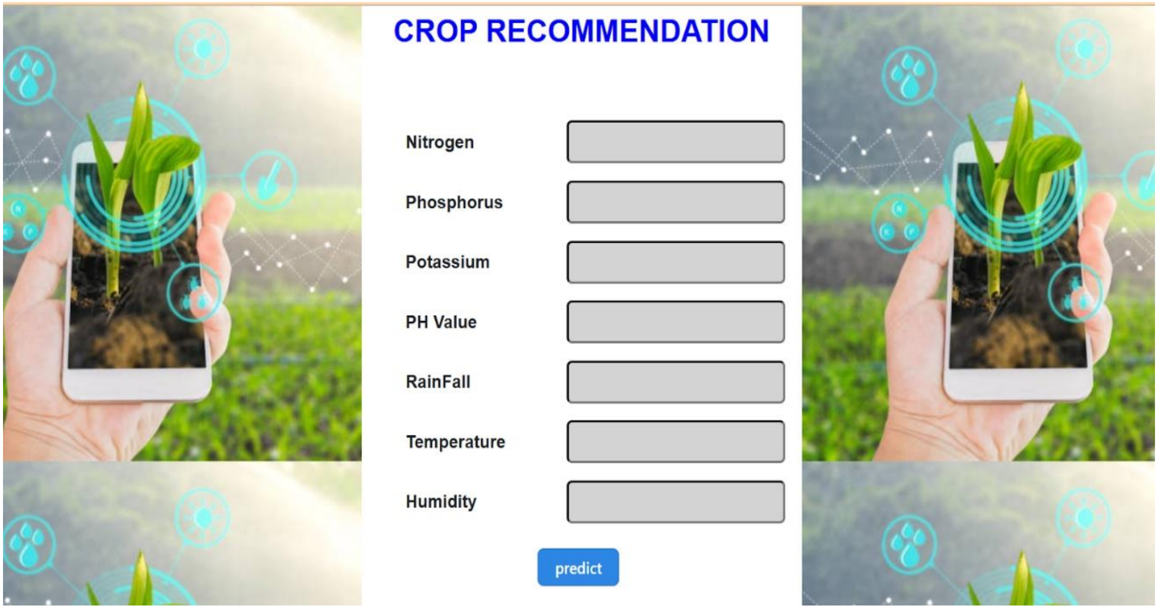

Figure 13: Data entering page

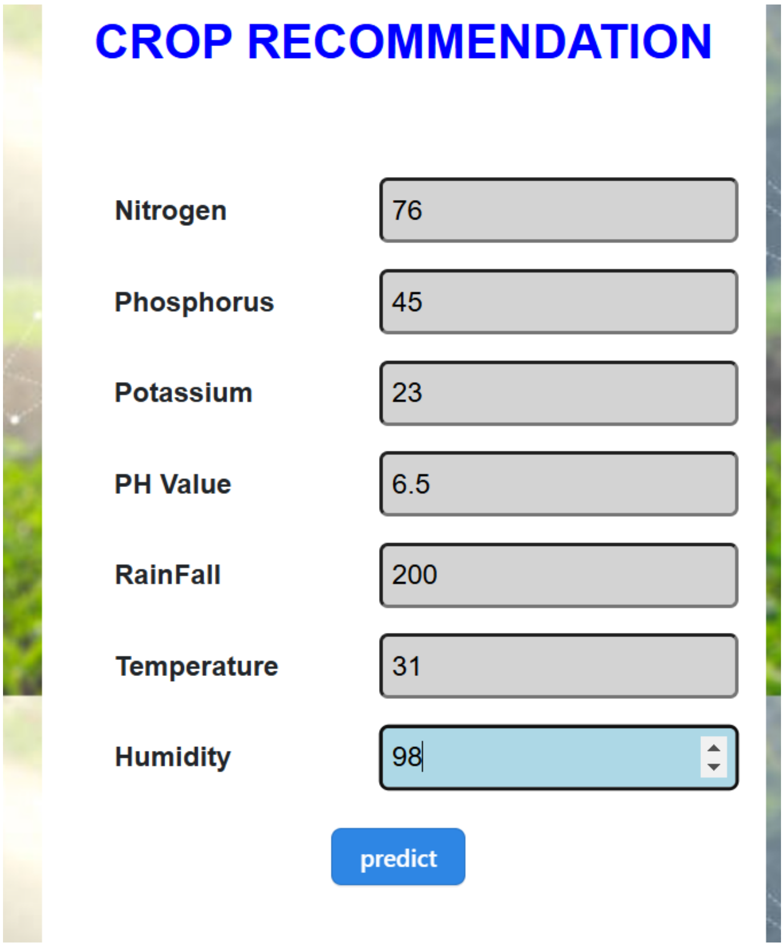

Figure 14: Submission page

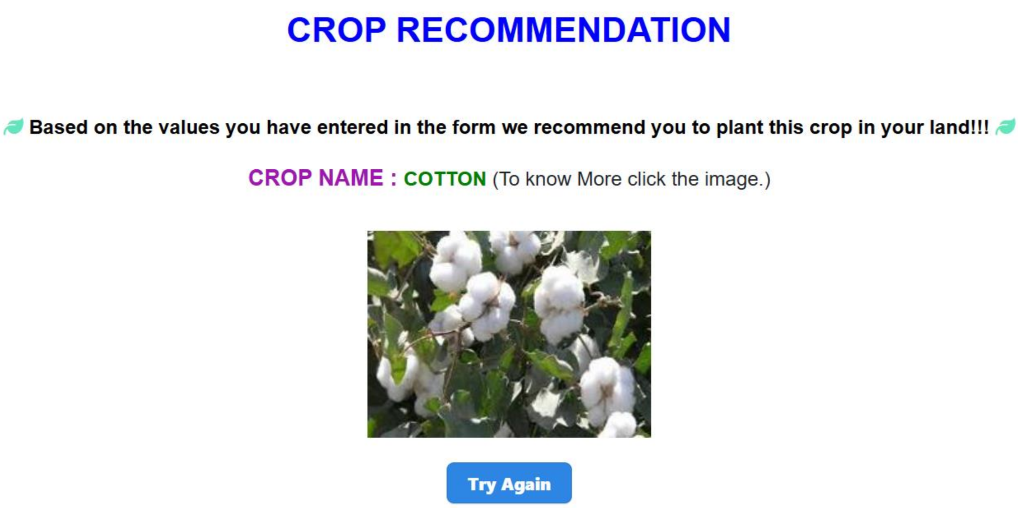

Figure 15: Crop recommendation page

REFERENCES

1. E Eswari, L. Vanitha, "Crop Yield Prediction in Tamil Nadu Using Bayesian Network", *International Journal of Intellectual Advancements and Research in Engineering Computations*, Volume-6, Issue-2, ISSN: 2348-2079.
2. K.E Eswari, L. Vanitha, "Crop Yield Prediction in Tamil Nadu Using Bayesian Network", *International Journal of Intellectual Advancements and Research in Engineering Computations*, Volume-6, Issue-2, ISSN: 2348-2079.
3. Anna Chlingaryana, Salah Sukkarieha, Brett Whelanb, "Machine Learning Approaches for Crop Yield Prediction and Nitrogen Status Estimation in Precision Agriculture: A Review", *Computers and Electronics in Agriculture*, 151 (2018) 61–69, Elisver, 2018.
4. Dakshayini Patil et al., "Rice Crop Yield Prediction using Data Mining Techniques: An Overview", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 7, May 2017.
5. Jharna Majumdar, Sneha Naraseeyappa, and Shilpa Ankalaki, "Analysis of Agriculture Data Using Data Mining Techniques: Application of Big Data", *Springer Journal*, 2017.
6. K. Menaka, N. Yuvaraj, "Indian Journal of Innovations and Developments", Vol 5(12), October 2016, ISSN (online): 2277-5390, ISSN (Print): 2277-5382.
7. Snehal S. Dahikar, Dr. Sandeep V. Rode, "Agricultural Crop Yield Prediction Using Artificial Neural Network Approach", *International Journal of Innovative and Research in Electrical, Instrumentation and Control Engineering*, Volume 2, Issue 2, 2014.
8. D. Ramesh, B. Vishnu Vardhan, "Analysis of Crop Yield Prediction Using Data Mining Techniques", *International Journal of Research in Engineering and Technology*, eISSN: 2319-1163.
9. Ramesh A. Medar, "A Survey on Data Mining Techniques for Crop Yield Prediction", *International Journal of Advance in Computer Science and Management Studies*, ISSN: 2231-7782, Volume 2, Issue 9, 2014.
10. Pooja M C et al., "Implementation of Crop Yield Forecasting Using Data Mining", *International Research Journal of Engineering and Technology*.