A

Mini Project

On

CROP RECOMMENDATION USING RANDOM FOREST ML ALGORITHM

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

K.VARSHINI (217R1A05G1)

J.POOJITHA (217R1A05F0)

D.DAYANIDHI (217R1A05E4)

Under the Guidance of

SABA SULTANA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2021-25

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
TECHNICAL CAMPUS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "**CROP RECOMMENDATION USING RANDOM FOREST ML ALGORITHM**" being submitted by **K.VARSHINI (217R1A05G1), J.POOJITHA (217R1A05F0) & D.DAYANIDHI (217R1A05E4)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University,Hyderabad is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

SABA SULTANA **Assistant Professor** **INTERNAL GUIDE**	**Dr. A. RAJI REDDY** **DIRECTOR**
Dr. N.BHASKAR **HoD**	**EXTERNAL EXAMINER**

Submitted for viva voice Examination held on ________________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express our profound gratitude and deep regard to my guide

SABA SULTANA , Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. K.Maheswari, Dr. J.Narasimha rao, Ms. K.Shilpa, Mr. K.Ranjith Reddy,** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. N .Bhaskar,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this project would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

K.VARSHINI(217R1A05G1)

J.POOJITHA(217R1A05F0)

D.DAYANIDHI(217R1A05E4)

ABSTRACT

Agriculture and its allied sectors are undoubtedly the largest providers of livelihoods in rural India. The agriculture sector is also a significant contributor factor to the country’s Gross Domestic Product (GDP). Blessing to the country is the overwhelming size of the agricultural sector. However, regrettable is the yield per hectare of crops in comparison to international standards. This is one of the possible causes for a higher suicide rate among marginal farmers in India. This paper proposes a viable and user-friendly yield prediction system for the farmers. The proposed system provides connectivity to farmers via a mobile application. GPS helps to identify the user location. The user provides the area & soil type as input. Machine learning algorithms allow choosing the most profitable crop list or predicting the crop yield for a user-selected crop. To predict the crop yield, selected Machine Learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), Multivariate Linear Regression (MLR), and K-Nearest Neighbour (KNN) are used. Among them, the Random Forest showed the best results with 95% accuracy. Additionally, the system also suggests the best time to use the fertilizers to boost up the yield.

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

TABLE OF CONTENTS

1.INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as "Crop Recommendation Using Random Forest Machine Learning Algorithm." This software helps farmers or users input crop-related parameters and receive recommendations on the most suitable crop to grow. The project leverages machine learning techniques to analyze various factors like soil type, temperature, rainfall, and pH levels. Using the Random Forest algorithm, it identifies patterns from the data and suggests optimal crops based on environmental conditions, aiming to improve agricultural productivity.The scope of this project extends beyond providing crop recommendations. It can serve as an intelligent advisory tool that contributes to sustainable farming practices by optimizing the use of resources such as water, fertilizers, and pesticides. This system can be implemented across various regions, enabling farmers to adopt smart farming techniques tailored to their local conditions. Additionally, the project can evolve by incorporating new features such as real-time data inputs through IoT sensors, weather forecasting, and market price analysis, further enriching the decision-making process for farmers.

The proposed system aims to address the challenges faced by the agricultural community, including the unpredictability of weather, soil variability, and the lack of access to expert advice. It offers a scalable, efficient, and easy-to-use solution that can be accessed via web or mobile platforms. The modular architecture of the system ensures smooth data flow, easy integration of new features, and seamless updates to improve the model over time. The project's ultimate objective is to empower farmers, reduce agricultural risks, and promote sustainable farming practices, thereby contributing to increased food production and economic growth.

1.2 PROJECT PURPOSE

This project has been developed to assist farmers and agricultural professionals in making informed crop selection decisions based on environmental conditions. The system analyzes input data such as soil type, temperature, rainfall, and pH levels to recommend the most suitable crops. It ensures accurate, data-driven predictions using the Random Forest algorithm, which improves the chances of higher yield and sustainable farming. The goal is to optimize resource utilization and enhance productivity by providing reliable recommendations aligned with local conditions. Another significant goal of the project is to reduce the uncertainty caused by climate change and unpredictable weather patterns. The system encourages farmers to make proactive decisions by providing recommendations based on current data trends. Additionally, the project aims to enhance the economic stability of farmers by guiding them towards crops with better market potential and higher profitability, reducing the chance of financial losses.

Ultimately, this project seeks to foster sustainable agricultural practices by integrating technology with farming. It ensures that the agricultural community benefits from advancements in machine learning, making expert recommendations accessible to a wider audience. The project contributes towards improving food security, supporting rural economies, and promoting smart farming practices that are vital for long-term agricultural development.

1.3 PROJECT FEATURES

The main features of this project include providing crop recommendations by analyzing environmental data such as soil type, temperature, rainfall, and pH levels. The Random Forest algorithm processes this data to offer the most suitable crop suggestions. The system functions as a decision support tool, helping users overcome challenges related to crop selection. It presents recommendations based on historical data and patterns. Users can review and modify input parameters if needed, and the system will update recommendations accordingly. This iterative process ensures that users receive the most optimized crop suggestions tailored to their specific conditions. The project also offers the flexibility to include additional features like weather forecasting integration, helping farmers make informed decisions by factoring in future weather patterns. Market price analysis can be incorporated to recommend crops that are not only suitable for the conditions but also profitable, guiding farmers toward better financial outcomes. Another feature is the ability to store historical data in the system's database, allowing users to review past recommendations, crop performance, and outcomes to refine future decisions.

Furthermore, the system promotes sustainability by optimizing resource usage through precision agriculture. Future versions could provide suggestions for appropriate fertilizers, pesticides, and irrigation levels, helping farmers reduce waste and minimize environmental harm. The back-end is designed for scalability, supporting multiple users simultaneously, and allows administrators to update datasets and improve the model periodically with new agricultural data. These features collectively ensure that the project delivers practical, sustainable, and impactful solutions to address the evolving challenges in agriculture.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System analysis is a crucial phase in the development of the "Crop Recommendation Using Random Forest Machine Learning Algorithm" project. This phase involves a detailed study of agricultural practices and environmental factors that affect crop selection. The system is examined to identify the key inputs, such as soil type, pH levels, temperature, and rainfall, and how these elements interact to influence crop productivity. The role of the system analyst is to thoroughly investigate existing farming challenges and data sources, ensuring that the algorithm can address these effectively. The primary question explored is, "How can we recommend the most suitable crop for specific conditions?" Once the analysis is complete, there is a clear understanding of how to structure the inputs, processes, and outputs to achieve the project's objective.

The system is analyzed to work efficiently within an interconnected architecture, where user input, algorithm processing, and result generation function cohesively. The Random Forest algorithm is selected after evaluating various machine learning models due to its robustness and ability to handle large datasets with high accuracy. The system must also meet performance criteria, ensuring that it provides predictions in a timely manner, even with complex data inputs. During the analysis, factors such as scalability, modularity, and future feature upgrades—like IoT integration and weather forecasting—are considered to ensure the system can evolve with technological advancements. Additionally, the analyst identifies potential risks, such as data inconsistencies or inaccurate predictions, and ensures appropriate mitigation strategies are in place. This comprehensive analysis ensures that the final product will be user-friendly, adaptable, and capable of addressing the challenges faced in modern agriculture.

2.1 PROBLEM DEFINITION

A detailed study of the crop selection process is essential to understand the challenges faced by farmers, such as poor yield, improper crop selection, and inefficient resource utilization. Various factors like soil quality, rainfall, temperature, and pH levels must be analyzed to determine their impact on crop performance. In many cases, farmers rely on traditional knowledge, which may not always align with current environmental conditions, leading to suboptimal crop choices. This system aims to solve these issues by leveraging machine learning techniques, specifically the Random Forest algorithm, to provide data-driven crop recommendations. The existing farming practices are closely examined, and the gaps identified are addressed through this solution. The recommendations are continuously refined based on user feedback to ensure they meet the specific needs of farmers. This iterative approach ensures that the final system offers reliable and practical crop suggestions.

This project addresses these challenges by developing a system that uses the **Random Forest Machine Learning Algorithm** to recommend crops based on multiple parameters, such as soil type, pH levels, temperature, and rainfall. The aim is to reduce the guesswork involved in crop selection by providing precise, data-driven recommendations tailored to specific conditions. By automating the decision-making process, the system minimizes the need for manual calculations and reduces the impact of human error. Additionally, it helps promote sustainable agriculture by optimizing resource usage and improving crop yields. The problem is framed not only as an agricultural challenge but also as a technical one—how to develop an accurate and user-friendly system capable of guiding farmers toward better decisions, even in the face of uncertain environmental conditions.

2.2 EXISTING SYSTEM

In the existing system, crop selection primarily relies on traditional farming knowledge, intuition, or generalized guidelines that may not account for specific environmental conditions. Some systems use basic statistical models or linear algorithms for recommendations, but these lack the sophistication to handle complex, non-linear relationships between factors like soil type, pH levels, rainfall, and temperature. As a result, the accuracy and relevance of these recommendations are limited, leading to suboptimal crop choices. Moreover, existing tools may not dynamically update based on changing environmental conditions, reducing their practical utility for farmers.

Current digital systems that attempt to provide crop recommendations usually rely on basic algorithms or static rule-based models, which do not perform well when dealing with large datasets or complex factors like weather variability. Additionally, these systems may not support continuous learning or adaptability, limiting their usefulness over time as conditions evolve. Some platforms are also limited in their accessibility, requiring high-end hardware or internet access, which can be a challenge for farmers in remote areas. The absence of seamless integration with other agricultural tools—such as weather forecasting services or IoT sensors for real-time soil monitoring—further limits the effectiveness of these existing systems. Therefore, there is a pressing need for a more robust, scalable, and accessible system that can provide dynamic, accurate crop recommendations while being easy to use for farmers.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

The existing systems for crop recommendation face several challenges:

- More Classification Required
- Time-Consuming
- Requires Manual Calculations

To overcome these limitations and improve accuracy, a more efficient system needs to be implemented using advanced machine learning techniques.

2.3 PROPOSED SYSTEM

The aim of the proposed system is to develop an advanced crop recommendation system using the Random Forest algorithm. This system will overcome the limitations of existing systems by providing more precise recommendations based on complex environmental factors. It integrates real-time data inputs, such as weather and soil conditions, to ensure relevant crop suggestions. The proposed system reduces manual effort, automates calculations, and delivers recommendations quickly and accurately. It offers a user-friendly interface that allows farmers to interact easily and receive timely crop recommendations without delays. In addition to crop recommendations, the system is designed to be flexible and scalable, allowing it to integrate future functionalities such as weather forecasts and market price analysis. These enhancements would not only ensure that the recommended crops are aligned with environmental conditions but also help farmers make decisions based on economic viability. The system can also be expanded to offer personalized advice on irrigation levels, fertilizer requirements, and pest control, promoting sustainable agricultural practices. This ensures that resources are used optimally, reducing waste and environmental impact.

The user interface is designed to be intuitive and accessible, enabling farmers to input data easily and obtain results quickly, even with minimal technical expertise. The modular architecture of the system supports smooth updates and future enhancements, such as the incorporation of IoT devices for real-time data collection. By using machine learning, the model will continuously improve over time, becoming more precise as new data is fed into the system. The proposed system not only empowers farmers to make informed decisions but also plays a role in promoting sustainable agriculture, supporting food security, and enhancing rural livelihoods.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system offers several benefits:

- High Accuracy: It provides precise crop recommendations by analyzing multiple environmental factors.
- Real-Time Data Integration: Adapts to changing conditions, offering dynamic recommendations.
- Reduced Time Consumption: Automates processes, delivering results faster.
- Improved Efficiency: Requires minimal manual input, reducing errors.
- User-Friendly Interface: Easy to use for farmers with little technical knowledge.
- Broad Compatibility: Works on various devices and configurations with low resource requirements.

2.4 FEASIBILITY STUDY

The feasibility of the "Crop Recommendation Using Random Forest Machine Learning Algorithm" project is analyzed in this phase, and a general plan is proposed along with estimated costs and benefits. The objective is to ensure that the system will not become a burden to the users or stakeholders. The following three key considerations are involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility

- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

Economic feasibility evaluates the cost-effectiveness of the project. The system is being developed using open-source tools like Python and Scikit-learn, eliminating licensing fees. Hardware requirements such as an i3 processor and 4GB RAM are modest, minimizing infrastructure costs. Since the project is a part of academic or research work, labor and development costs are significantly reduced. The expected benefits include higher crop yield, optimized resource usage, and reduced crop failure risks, which outweigh the minimal investment. Therefore, the project is economically feasible.

The system is designed to function efficiently on basic hardware, making it accessible to users even in low-resource settings, further ensuring cost-effectiveness. Once implemented, the system lowers the need for expert consultations or trial-and-error farming methods, reducing operational costs for farmers. Moreover, by minimizing resource wastage and improving crop selection, farmers can achieve higher productivity with fewer inputs, translating into long-term economic savings. As the system supports scalable deployment, it can be rolled out to multiple regions without significant additional investment, making it a sustainable solution with a high potential for positive economic impact at both local and national levels. Thus, the economic feasibility of the project is validated by its ability to deliver cost-efficient solutions and improve agricultural profitability over time.

2.4.2 TECHNICAL FEASIBILITY

Technical feasibility assesses whether the existing technical resources are adequate for the development and operation of the system. The project leverages the Random Forest algorithm, which can efficiently run on systems with basic specifications, such as an Intel i3 processor and 4GB RAM. Software tools like Python and Jupyter Notebook are open-source, making them easily accessible. Minimal changes are required for implementation, ensuring compatibility with most agricultural data sources. Therefore, the project is technically feasible.

The system has minimal technical barriers for users, as it offers a user-friendly interface for farmers and agricultural professionals, even with limited technological expertise. Automated processes within the system reduce the need for manual intervention, streamlining operations and minimizing errors. With proper training and initial guidance, users can effectively operate the platform, ensuring its long-term sustainability. Therefore, the project is not only technically feasible but also positioned to adapt to future developments in agriculture and technology.

2.4.3 SOCIAL FEASIBILITY

Social feasibility evaluates the acceptance and impact of the proposed system on users. The project aims to assist farmers by providing data-driven crop recommendations, which directly contributes to improved agricultural practices. The user-vsvsfriendly design ensures that farmers can easily interact with the system, even with limited technical knowledge. Additionally, the system encourages sustainable farming by promoting optimized crop selection, benefiting society as a whole. Therefore, the project is socially feasible and expected to receive positive user acceptance.

The system also promotes social welfare by supporting sustainable farming practices, encouraging optimal use of resources like water and fertilizers, which benefits both farmers and the environment. With features that reduce waste and enhance productivity, the project contributes to long-term agricultural sustainability and food security. Additionally, by reducing crop failures and financial losses, the project helps farmers achieve greater economic resilience, which is crucial for their well-being and the stability of agricultural communities. As farmers experience success with this technology, it can foster a culture of innovation in agriculture, encouraging the adoption of smart farming techniques across communities. This ripple effect can contribute to rural development, improving the social fabric and overall quality of life in farming regions.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware requirements specify the necessary physical components needed for the effective operation of the crop recommendation system. The following are some hardware requirements:

- Processor: Intel i3 or above.
- RAM: 4 GB or above.
- Hard Disk: 40 GB of available storage.

2.5.2 SOFTWARE REQUIREMENTS:

Software requirements outline the necessary software components and logical characteristics of the system. The following are some software requirements:

- Operating System: Windows 8 or above.
- Programming Language: Python.

3.ARCHITECTURE

3. ARCHITECTURE

3.1 SYSTEM ARCITECTURE

The system architecture for the project follows a modular design to ensure smooth data flow from input to output, with efficient processing using machine learning models. Below is the architecture:

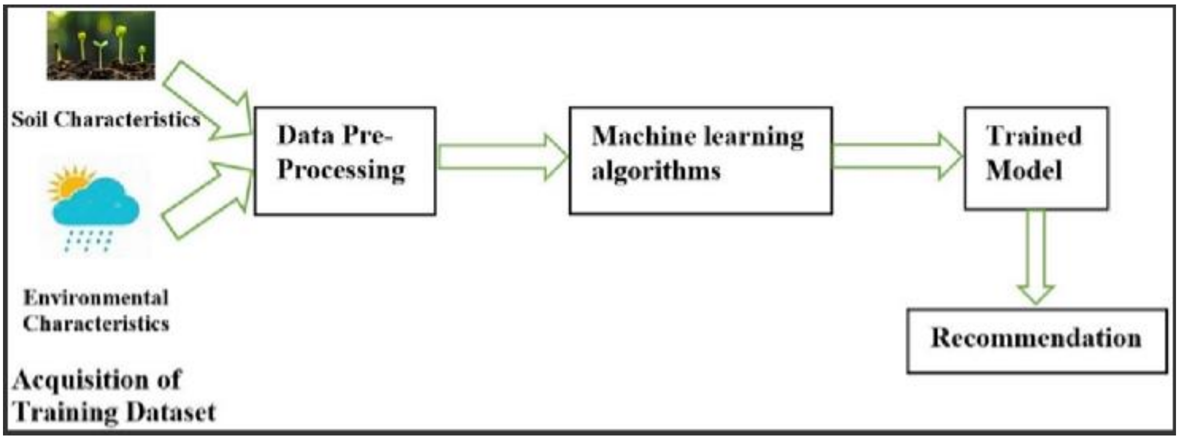


Figure 3.1: System Architecture of Crop Recommendation

3.2 DESCRIPTION

User Interface: The user interface acts as the front-end portal where farmers or agricultural users input environmental parameters, such as soil type, temperature, rainfall, and pH levels. This interface can be web-based or a desktop application, offering ease of use even to those with minimal technical knowledge. It collects input data and displays crop recommendations generated by the system.

Database: The database stores critical information such as historical data, crop details, environmental parameters, and prediction results. It serves as a backend repository to manage both input and output data, enabling future analysis and model retraining. A lightweight database like SQLite is used for efficient data storage and retrieval.

Processing Server: The processing server handles all interactions between the user interface, the machine learning model, and the database. It processes user inputs, runs the machine learning predictions, retrieves relevant data from the database, and delivers recommendations back to the user. The server ensures smooth communication between all components and ensures quick response times.

3.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. In the use case diagram we have basically two actors who are the user and the administrator. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

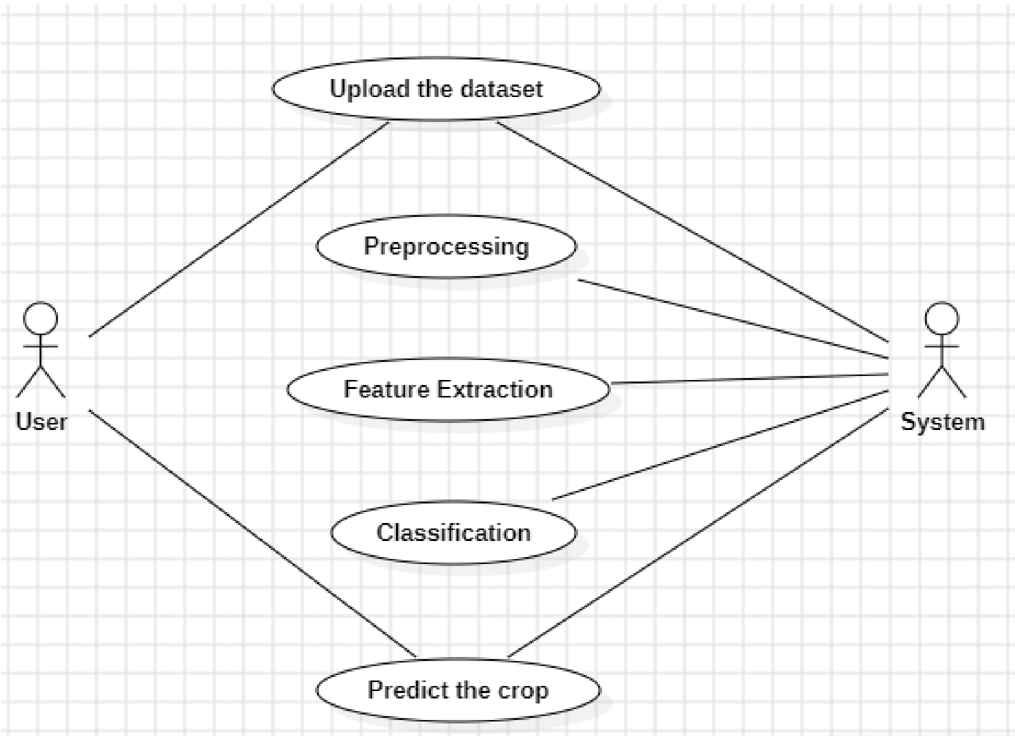


Figure 3.2: Use Case Diagram of Crop Recommendation

3.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

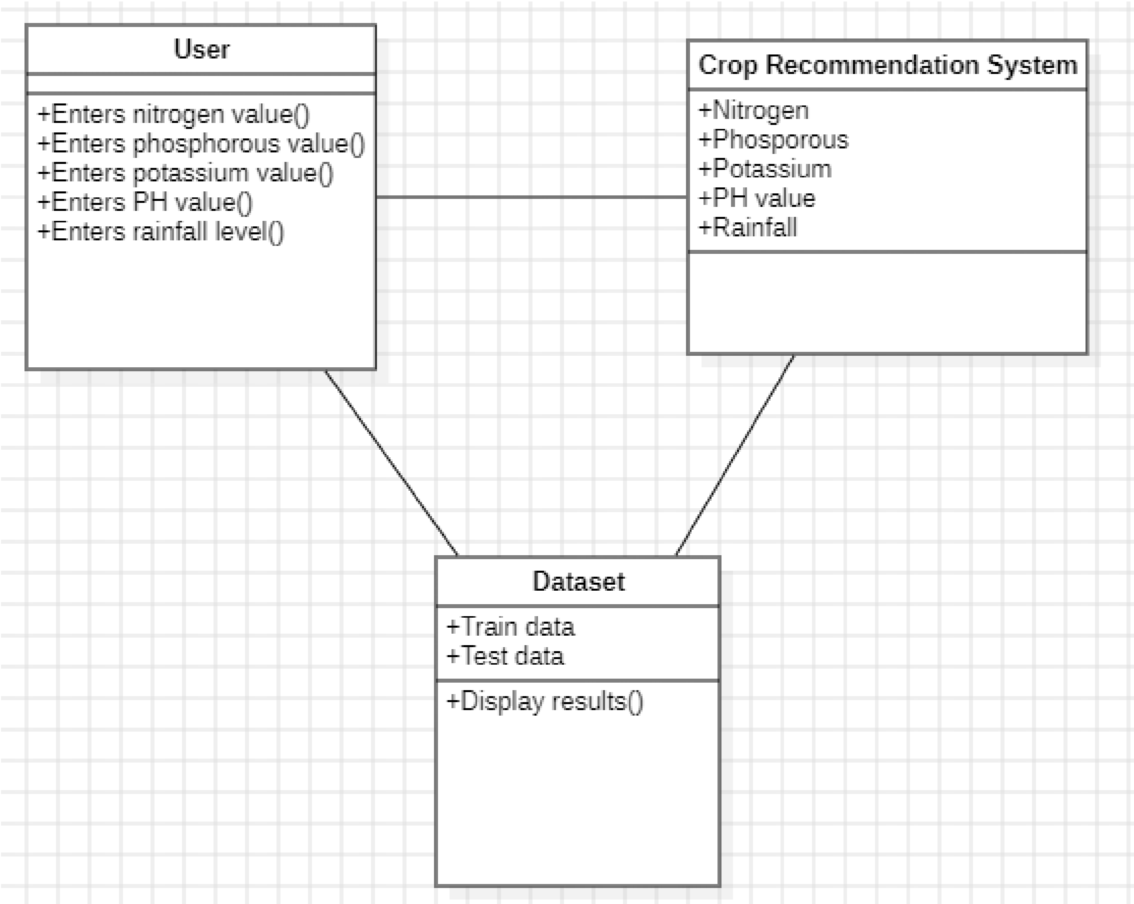


Figure 3.3: Class Diagram for Crop Recommendation

3.5 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that details how operations are carried out. It is an essential component used in processes related to analysis, design and documentation.

Sequence diagrams are organized according to time, with the time progressing as you go down the page. Object interactions usually begin at the top of a diagram and end at the bottom.

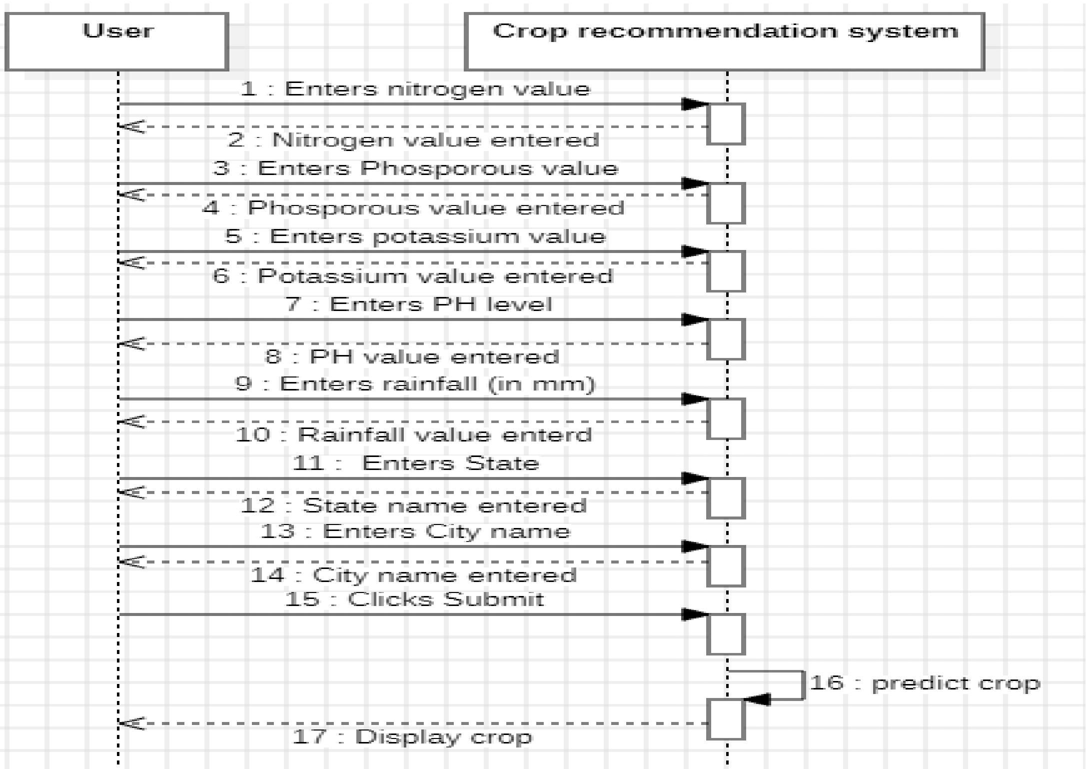


Figure 3.4: Sequence Diagram of crop recommendation

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

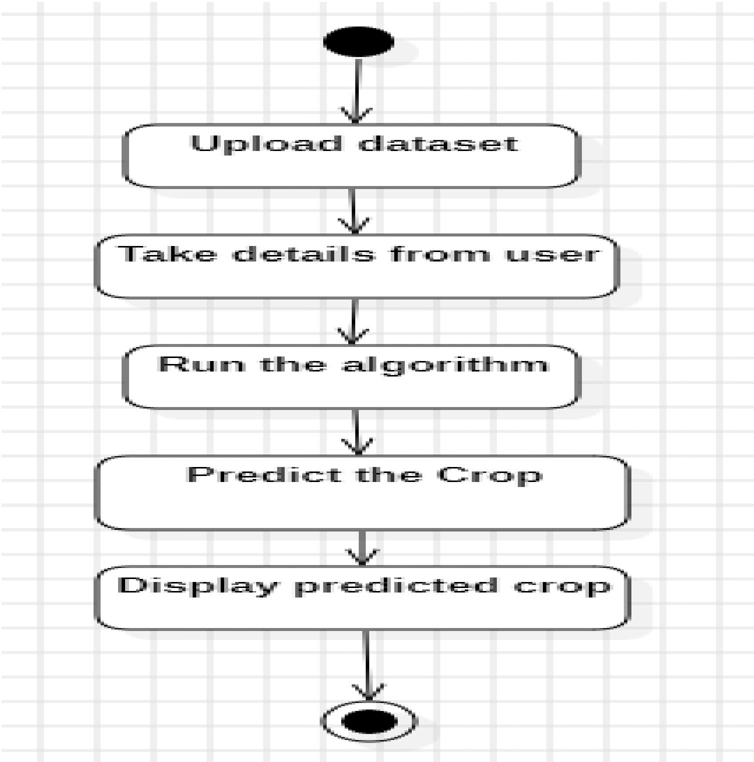


Figure 3.5: Activity Diagram of Crop Recommendation

4.IMPLEMENTATION

4.IMPLEMENTATION

4.1 SOURCE CODE

app.py:

```
# Importing essential libraries and modules

from flask import Flask, render_template, request, Markup
import numpy as np
import pandas as pd
import requests
import pickle
import io

# Loading crop recommendation model

crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
    open(crop_recommendation_model_path, 'rb'))

app = Flask(__name__)

# render home page

@ app.route('/index')
def index():
    title = 'CropRecommendation'
    return render_template('index.html', title=title)

@ app.route('/')
def home():
    title = 'CropRecommendation'
    return render_template('index.html', title=title)

# render crop recommendation form page

@ app.route('/crop-recommend')
def crop_recommend():
    title = 'Harvestify - Crop Recommendation'
    return render_template('crop.html', title=title)

# render fertilizer recommendation form page

@ app.route('/fertilizer')
```

```
def fertilizer_recommendation():
    title = 'Harvestify - Fertilizer Suggestion'

    return render_template('fertilizer.html', title=title)

# render disease prediction input page

# RENDER PREDICTION PAGES

# render crop recommendation result page

@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'Harvestify - Crop Recommendation'

    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        #if weather_fetch(city) != None:
            #temperature, humidity = weather_fetch(city)
        data = np.array([[N, P, K, (N+P), (N+K), ph, rainfall]])
        my_prediction = crop_recommendation_model.predict(data)
        final_prediction = my_prediction[0]
        return render_template('crop-result.html', prediction=final_prediction, title=title)

# render fertilizer recommendation result page

@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'Harvestify - Fertilizer Suggestion'

    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['pottasium'])
    # ph = float(request.form['ph'])

    df = pd.read_csv('Data/fertilizer.csv')

    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
```

```
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
    elif max_value == "P":
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"

    response = Markup(str(fertilizer_dic[key]))

    return render_template('fertilizer-result.html', recommendation=response, title=title)

# render disease prediction result page

@app.route('/disease-predict', methods=['GET', 'POST'])def disease_prediction():
        title = 'Harvestify - Disease Detection

    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files.get('file')
        if not file:
            return render_template('disease.html', title=title)
        try:
            img = file.read()

            prediction = predict_image(img)

            prediction = Markup(str(disease_dic[prediction]))
            return render_template('disease-result.html', prediction=prediction, title=title)
        except:
            pass
    return render_template('disease.html', title=title)
```

```
if __name__ == '__main__':
    app.run(debug=False)
```

5.RESULTS

5. RESULTS

N	P	K	temperatu	humidity	ph	rainfall	label
90	42	43	20.87974	82.00274	6.502985	202.9355	rice
85	58	41	21.77046	80.31964	7.038096	226.6555	rice
60	55	44	23.00446	82.32076	7.840207	263.9642	rice
74	35	40	26.4911	80.15836	6.980401	242.864	rice
78	42	42	20.13017	81.60487	7.628473	262.7173	rice
69	37	42	23.05805	83.37012	7.073454	251.055	rice
69	55	38	22.70884	82.63941	5.700806	271.3249	rice
94	53	40	20.27774	82.89409	5.718627	241.9742	rice
89	54	38	24.51588	83.53522	6.685346	230.4462	rice
68	58	38	23.22397	83.03323	6.336254	221.2092	rice
91	53	40	26.52724	81.41754	5.386168	264.6149	rice
90	46	42	23.97898	81.45062	7.502834	250.0832	rice
78	58	44	26.8008	80.88685	5.108682	284.4365	rice
93	56	36	24.01498	82.05687	6.984354	185.2773	rice
94	50	37	25.66585	80.66385	6.94802	209.587	rice
60	48	39	24.28209	80.30026	7.042299	231.0863	rice
85	38	41	21.58712	82.78837	6.249051	276.6552	rice
91	35	39	23.79392	80.41818	6.97086	206.2612	rice
77	38	36	21.86525	80.1923	5.953933	224.555	rice
88	35	40	23.57944	83.5876	5.853932	291.2987	rice
89	45	36	21.32504	80.47476	6.442475	185.4975	rice
76	40	43	25.15746	83.11713	5.070176	231.3843	rice
67	59	41	21.94767	80.97384	6.012633	213.3561	rice
83	41	43	21.05254	82.6784	6.254028	233.1076	rice
98	47	37	23.48381	81.33265	7.375483	224.0581	rice
66	53	41	25.07564	80.52389	7.778915	257.0039	rice
97	59	43	26.35927	84.04404	6.2865	271.3586	rice
97	50	41	24.52923	80.54499	7.07096	260.2634	rice
60	49	44	20.77576	84.49774	6.244841	240.0811	rice

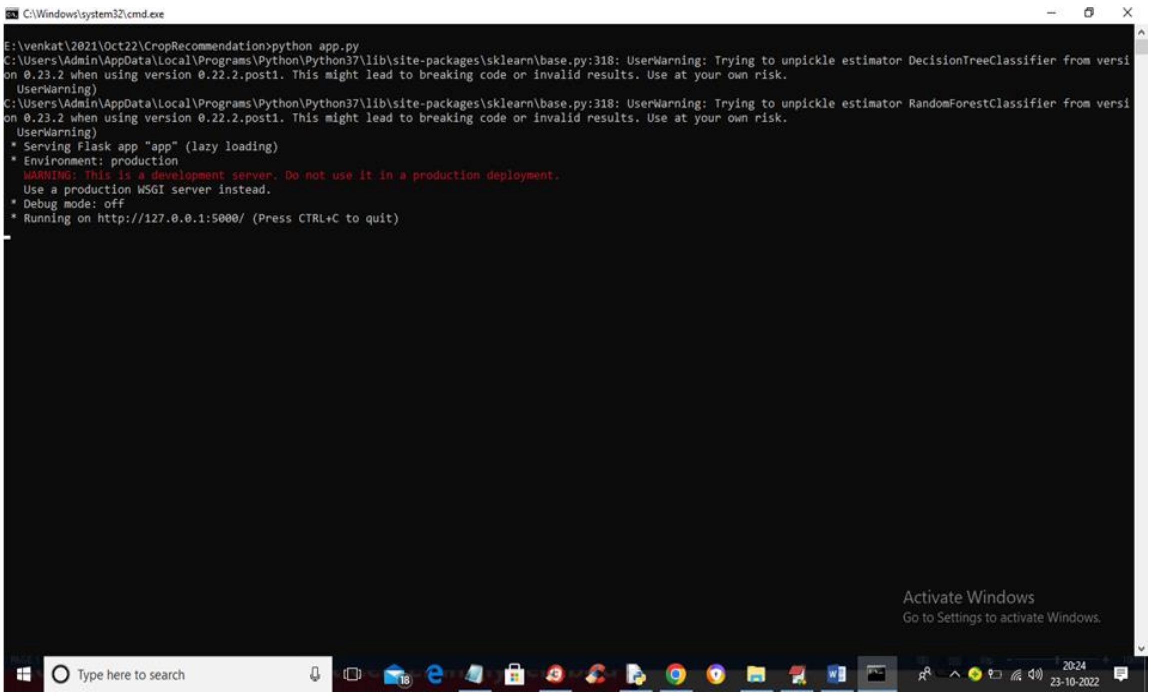
Screenshot 5.1: Training dataset for crop recommendation

The dataset contains input-output pairs, where the input features are variables relevant to the problem (such as soil type, temperature, and rainfall for crop prediction), and the output labels represent the target (e.g., crop type). The purpose of the training dataset is to enable the model to find correlations between the input features and the target labels by adjusting internal parameters, like weights, during the learning process. For supervised learning, the training dataset must include both labeled data and representative samples that cover a wide range of potential inputs to help the model generalize well to unseen data.

```
15,13,29,1.11,4.11, jharkand,chakardharpur
45,33,9,2.9,3.5,lakshadweep,bitra island
12,9,6,5.9,6.8,chandigarh,chandigarh
32,123,113,101,122,jammu kashmir,bandipur
```

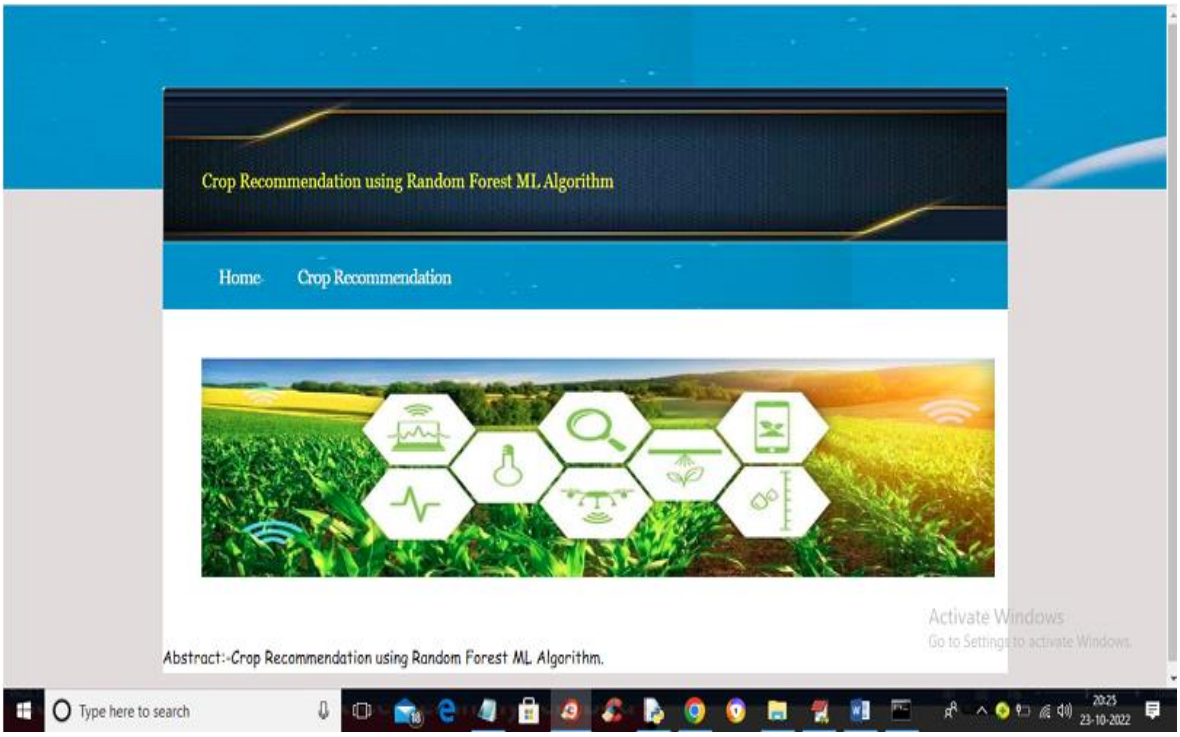
Screenshot 5.2: Testing dataset for crop recommendation

The testing dataset is essential for computing key performance metrics such as accuracy, precision, recall, F1 score, and mean squared error (MSE). These metrics help determine the effectiveness of the model and highlight potential weaknesses. In some cases, cross-validation techniques are employed to ensure the model's robustness across multiple testing subsets.



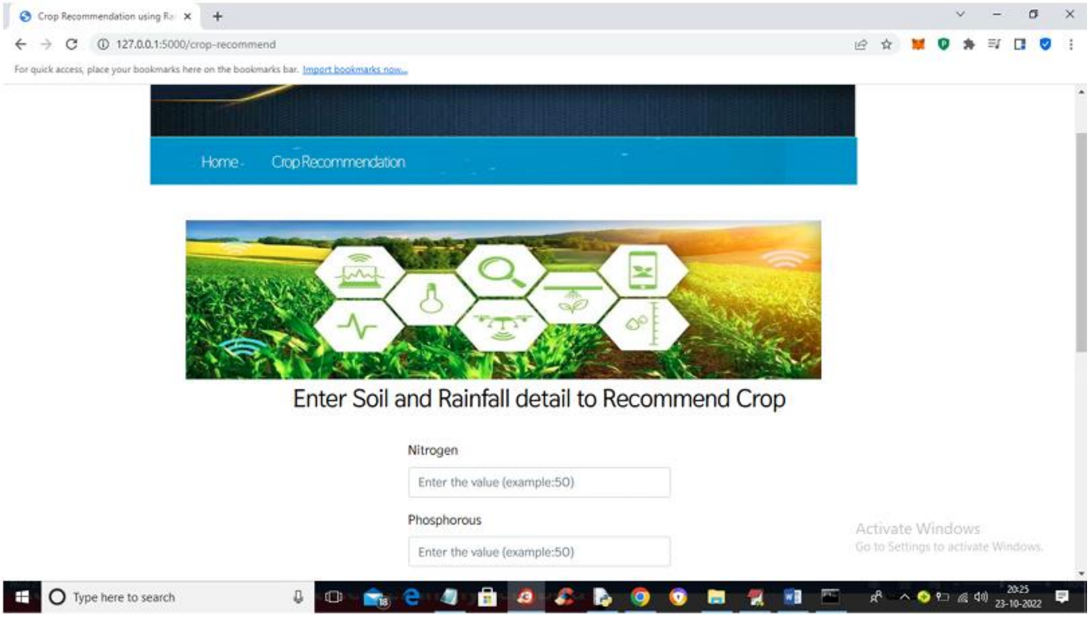
Screenshot 5.3: Open FLASK server and enter http://127.0.0.1:5000/index to get above screen

cmd.exe is the default command-line interpreter for Windows operating systems. It provides a text-based interface that allows users to execute commands, automate tasks, and interact with the underlying system. CMD.exe is an essential tool for developers, system administrators, and power users, enabling them to run scripts, configure system settings, manage files, and troubleshoot issues without a graphical interface.



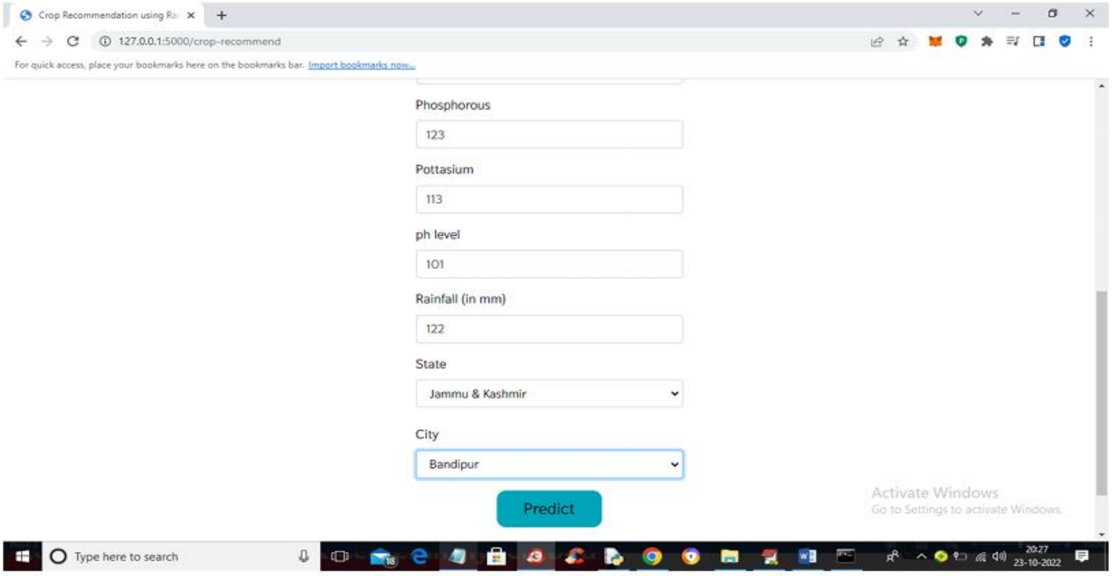
Screenshot 5.4: Home page for crop recommendation

In this page we have 2 buttons one navigates to home page and another button navigates to crop recommendation where the values are to be entered.



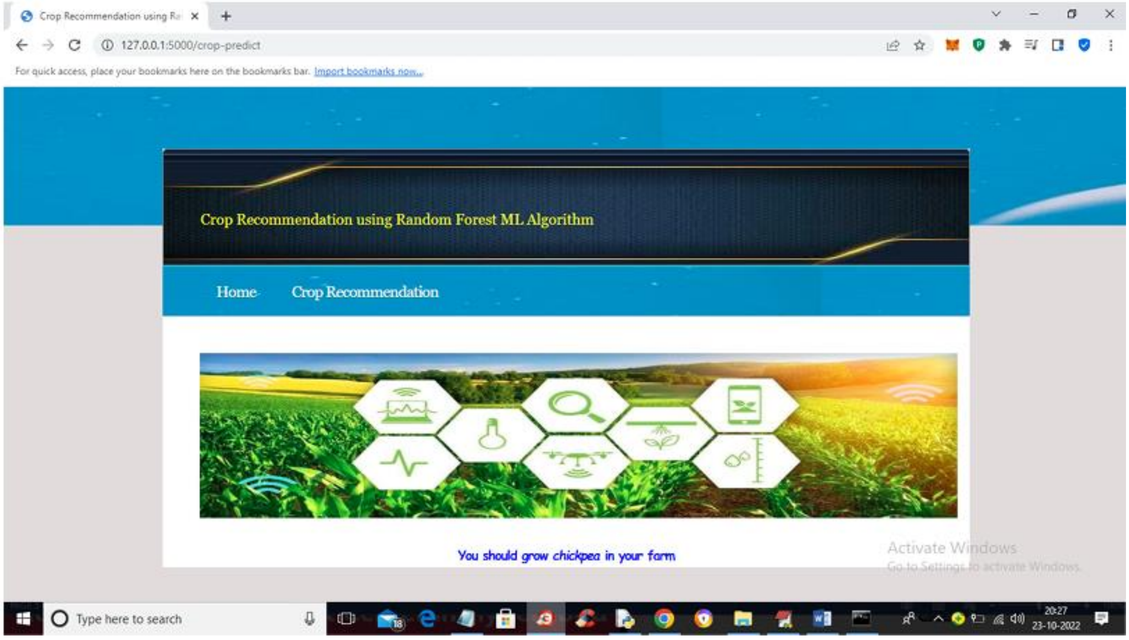
Screenshot 5.5: Enter soil rainfall detail to recommend crop

To use the crop recommendation system, users need to enter specific input values related to environmental and soil conditions. These inputs typically include parameters such as soil type, pH level, temperature, humidity, rainfall, and sunlight exposure. The accuracy of the recommendations depends on the quality of the data provided, as the machine learning model analyzes these inputs to suggest the most suitable crops for the given conditions. Once the values are entered correctly, the system processes them and generates a list of crops that are optimal for cultivation, helping farmers make informed decisions and achieve better yields.



Screenshot 5.6: Enter soil rainfall details to recommend crop

Additionally, the system ensures user-friendly input validation to avoid errors and improve prediction accuracy. If any value is out of range or missing, the system prompts the user to correct it, ensuring the data entered aligns with realistic agricultural conditions. This helps maintain the integrity of the recommendations. Once all required parameters are submitted, the machine learning model processes the inputs in real time and delivers crop suggestions tailored to the specific environment, empowering farmers and agricultural planners to optimize their resources and enhance productivity.



Screenshot 5.7: The system predicts the most suitable crop

The prediction page is designed to be intuitive and user-friendly, presenting the output in a clear and organized manner. In addition to the recommended crops, it may also provide additional information such as the predicted success rate, ideal growing conditions, and tips for cultivation. If any parameters are missing or incorrect, the page will display validation messages, prompting users to adjust the input values. This ensures the system delivers accurate and meaningful results to help farmers and agricultural planners make informed decisions.

6.TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Machine learning testing involves several key stages, such as model evaluation using metrics like accuracy, precision, recall, and F1 score, along with cross-validation techniques to reduce overfitting. These metrics help determine whether the model achieves the desired performance across various data subsets. Another critical part of testing is splitting data into training, validation, and test sets to ensure that the model can generalize well. The testing phase also includes identifying edge cases where the model might fail and assessing the impact of noise or missing data on predictions.

In addition to performance evaluation, testing in machine learning ensures that the models are fair, unbiased, and interpretable. As machine learning is increasingly used in real-world applications—such as healthcare, finance, and agriculture—it becomes essential to test models for potential biases that may arise from imbalanced datasets. Robust testing frameworks and automated tools, such as scikit-learn, TensorFlow, and PyTest, further facilitate the process by enabling reproducibility and systematic validation. Ultimately, rigorous testing ensures that the machine learning models deliver accurate, reliable, and ethical predictions in practical use cases.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

In machine learning, unit tests are typically applied to data preprocessing tasks, such as checking whether missing values are handled correctly or ensuring that categorical data is encoded properly. It is also crucial to test feature engineering steps, ensuring that transformations like normalization or scaling are applied consistently. Moreover, unit tests are essential for model training functions to validate that the model receives input in the correct shape and format. For instance, ensuring that the Random Forest model can be trained without errors on a sample dataset is an important step.

Testing evaluation metrics is another critical aspect of unit testing in machine learning. For example, developers need to confirm that accuracy, precision, or F1-score functions return expected values for given inputs. Automated unit testing frameworks like PyTest, unittest, or TensorFlow’s testing modules help streamline this process by enabling test case creation and validation. While unit testing does not directly measure the model's final performance on unseen data, it ensures that individual components function correctly, reducing the chance of failures and inconsistencies throughout the machine learning pipeline. This modular approach simplifies debugging and improves the maintainability and reliability of the system.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Integration testing is essential in scenarios where multiple external systems or APIs interact, such as fetching live weather data for crop prediction models or using cloud services to store training datasets. It ensures that each part of the system communicates correctly and handles errors gracefully. Automated testing tools, such as pytest with mocking libraries or

CI/CD pipelines with tools like Jenkins and GitHub Actions, are often used to perform integration testing efficiently.

This type of testing plays a crucial role in building robust machine learning systems by ensuring smooth collaboration between different modules, reducing the risk of runtime errors, and making it easier to pinpoint issues in complex workflows. Integration testing ensures that the overall system functions as intended, providing reliable predictions when deployed in real-world environments.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing in machine learning focuses on verifying that the system's components and overall behavior align with the specified requirements. It ensures that the machine learning model produces expected outputs for given inputs and performs all intended tasks correctly. The primary goal is to validate the system's core functionalities—such as data preprocessing, feature extraction, model predictions, and performance evaluation—against defined use cases without delving into the internal code structure.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processe

6.3 TEST CASES

6.3.1 UPLOADING DATA

Test case ID	**Test case name**	**Purpose**	**Test Case**	**Output**
1	User uploads information	Used for crop recommendation	User uploads the data such as N_2, K, P, $P_{H,}$ Rainfall, state,city	Gives the most suitable farm to grow
2	User uploads 2nd information	Used for crop recommendation	User uploads the data such as N_2, K, P, $P_{H,}$ Rainfall by changing the values without changing state and city	Gives the most suitable farm to grow
3	User uploads 3rd information	Used for crop recommendation	User uploads the data such as N_2, K, P, $P_{H,}$ Rainfall by changing the state and city without changing the values	Gives the most suitable farm to grow

7.CONCLUSION&
FUTURE SCOPE

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The “Crop Recommendation Using Random Forest Machine Learning Algorithm” project provides an efficient solution to assist farmers and agricultural professionals in selecting the most suitable crops based on environmental factors such as soil type, pH level, rainfall, and temperature. By leveraging the power of the Random Forest algorithm, the system ensures high accuracy in predictions, thereby reducing the risks associated with poor crop selection and enhancing agricultural productivity. The system’s user-friendly interface makes it accessible to individuals with minimal technical knowledge, while its database-driven architecture ensures efficient data management and future scalability. Furthermore, the integration of real-time data and continuous feedback from users enables the model to improve over time, adapting to changing environmental conditions. In conclusion, this project not only empowers farmers to make data-driven decisions but also contributes to sustainable farming practices by optimizing crop yields. It bridges the gap between technology and agriculture, promoting smart farming techniques that can have long-term positive impacts on food production and rural economies

In addition to improving agricultural efficiency, the system also promotes sustainable farming practices. By guiding farmers towards the most suitable crops, the project optimizes the use of resources such as water, fertilizers, and pesticides, reducing waste and environmental harm. The system’s modular design allows for easy updates and future integrations, such as weather forecasts or real-time sensor data, ensuring that the model remains relevant and adaptive to changing conditions. Furthermore, its user-friendly interface ensures accessibility for farmers with minimal technical expertise, making the technology inclusive and impactful even in rural areas.

The successful implementation of this project can contribute significantly to improving food security and supporting rural economies. With the ability to reduce crop failures and boost profitability, it helps farmers build economic resilience, ultimately improving their livelihoods. As agriculture becomes more reliant on data and technology, this system lays the groundwork for future innovations in smart farming. In conclusion, the "Crop Recommendation Using Random Forest Algorithm" project not only addresses immediate agricultural challenges but also paves the way for long-term sustainable growth in the farming sector, ensuring that farmers can thrive even in the face of changing environmental and economic conditions.

7.2 FUTURE SCOPE

The “Crop Recommendation Using Random Forest Machine Learning Algorithm” project holds immense potential for future enhancements. One key direction is the integration of IoT devices, where sensors installed in fields can provide real-time data on soil conditions and environmental factors, enabling more precise recommendations. Additionally, incorporating weather forecasting models would help predict climatic changes and mitigate risks associated with unexpected weather patterns. Developing a mobile application for the system can improve accessibility for farmers, particularly in rural areas, and offering multilingual support will make it user-friendly for people from diverse linguistic backgrounds.Another promising area is the inclusion of market data, such as crop prices and demand trends, which would allow farmers to make economically sound decisions. Cloud integration will improve scalability, enabling the system to handle large datasets and cater to a broader user base. Collaboration with government bodies and agricultural research institutions can also enhance the system’s data quality and foster large-scale adoption. With these improvements, the project will remain a vital tool in advancing smart farming practices, promoting sustainable agriculture, and supporting global food security.

The system could also evolve into a multi-lingual and mobile-friendly platform, making it more accessible to farmers across different regions and linguistic backgrounds. Future versions could include voice-based interactions to accommodate farmers with limited literacy or technical skills. Additionally, collaborative features could be added to allow farmers and agricultural experts to share insights and experiences, fostering a community-driven approach to smart agriculture. With continuous advancements in technology, the crop recommendation system can play a crucial role in building climate-resilient agriculture by helping farmers adapt to unpredictable weather patterns and contributing to food security on a broader scale.

The future scope also includes the integration of government schemes and agricultural subsidies within the platform, providing farmers with tailored information about available financial support for specific crops. This feature would encourage farmers to adopt recommended crops with confidence, knowing that additional resources and incentives are accessible. Furthermore, partnerships with agricultural institutions and research centers could enable the system to update its models with the latest agronomic data and practices. Over time, the platform could evolve into a comprehensive agricultural management system by offering additional features such as pest and disease identification, fertilizer recommendations, and irrigation planning. These enhancements would make the system an all-in-one solution, empowering farmers with tools to optimize their entire farming process from seed selection to harvest, thus ensuring long-term agricultural sustainability and economic growth.

8.BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

Here are some references for your project on drought prediction and water level analysis using deep learning and satellite imagery:

1. Umamaheswari S, Sreeram S, Kritika N, Prasanth DJ, “BIoT: Blockchain-based IoT for Agriculture”, 11th International Conference on Advanced Computing (ICoAC), 2019 Dec 18 (pp. 324-327). IEEE.
2. Jain A. “Analysis of growth and instability in the area, production, yield, and price of rice in India”, Journal of Social Change and Development, 2018;2:46-66.
3. Manjula E, Djodiltachoumy S, “A model for prediction of crop yield” International Journal of Computational Intelligence and Informatics, 2017 Mar;6(4):2349-6363.
4. Sagar BM, Cauvery NK., “Agriculture Data Analytics in Crop Yield Estimation: A Critical Review”, Indonesian Journal of Electrical Engineering and Computer
5. Science, 2018 Dec;12(3):1087-93.
6. Wolfert S, Ge L, Verdouw C, Bogaardt MJ, “Big data in smart farming– a review. Agricultural Systems”, 2017 May 1;153:69-80.
7. Jones JW, Antle JM, Basso B, Boote KJ, Conant RT, Foster I, Godfray HC, Herrero M, Howitt RE, Janssen S, Keating BA, “Toward a new generation of agricultural system data, models, and knowledge products: State of agricultural systems science. Agricultural systems”, 2017 Jul 1;155:269-88.
8. Johnson LK, Bloom JD, Dunning RD, Gunter CC, Boyette MD, Creamer NG, “Farmer harvest decisions and vegetable loss in primary production. Agricultural Systems”, 2019 Nov 1;176:102672.
9. Kumar R, Singh MP, Kumar P, Singh JP, “Crop Selection Method to maximize crop yield rate using a machine learning technique”, International conference on smart technologies and management for computing, communication, controls, energy, and materials (ICSTM), 2015 May 6 (pp. 138-145). IEEE.

10. Sriram Rakshith.K, Dr.Deepak.G, Rajesh M, Sudharshan K S, Vasanth S, Harish Kumar N, “A Survey on Crop Prediction using Machine Learning Approach”, In International Journal for Research in Applied Science & Engineering Technology (IJRASET), April 2019, pp(3231- 3234).
11. Veenadhari S, Misra B, Singh CD, “Machine learning approach for forecasting crop yield based on climatic parameters”, In 2014 International Conference on Computer Communication and Informatics, 2014 Jan 3 (pp. 1-5). IEEE.
12. Ghadge R, Kulkarni J, More P, Nene S, Priya RL, “Prediction of crop yield using machine learning”, Int. Res. J. Eng. Technol. (IRJET), 2018 Feb;5.
13. Priya P, Muthaiah U, Balamurugan M, “Predicting yield of the crop using machine learning algorithm”, International Journal of Engineering Sciences & Research Technology, 2018 Apr;7(1):1-7.
14. S. Pavani, Augusta Sophy Beulet P., “Heuristic Prediction of Crop Yield Using Machine Learning Technique”, International Journal of Engineering and Advanced Technology (IJEAT), December 2019, pp (135-138)

8.2 GITHUB LINK

https://github.com/dayanidhi46/CROP-RECOMMENDATION-USING-RANDOM-FOREST-ML-ALGORITHM