### A Report on

**Crop Recommendation System**

Submitted for partial fulfillment of award of

### BACHELOR OF TECHNOLOGY

degree

In

Computer Science & Engineering

By

|  |  |
| --- | --- |
| Aditi Jain | 2000300100012 |
| Akshay Singh | 2000301540006 |
| Vaibhav Chauhan | 2000301530063 |

## Dr. Tripti Sharma

HOD, Department of CSE(DS**)**



**INDERPRASTHA ENGINEERING COLLEGE, GHAZIABAD**

## Dr. A P J ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW

**2023-24**

**Certificate**

Certified that Aditi Jain (2000300100012), Akshay Singh (2000301540006), Vaibhav Chauhan (2000301530063) **have carried out the project work presented in this report entitled** “Crop Recommendation System” **for the award of** Bachelor of Technology **from Inderprastha Engineering College, Ghaziabad, under my supervision. The report embodies result of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.**

Dr. Tripti Sharma

Designation:

Address:

Date: 07/02/2024

**Acknowledgement**

*We take this opportunity to thank our teachers and friends who helped us throughout the project.*

*First and foremost, I would like to thank my guide for the project (****Dr. Tripti Sharma, HoD, Data Science****) for her valuable advice and time during development of project.*

Aditi Jain Akshay Singh

2000300100012 2000301540006

Vaibhav Chauhan 2000301530063

**Declaration**

*We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature: Signature:*

*Name: Aditi Jain Name: Akshay Singh*

*Roll No.: 2000300100012 Roll No.: 2000301540006*

*Date: 07/02/2024 Date: 07/02/2024*

*Signature:*

*Name: Vaibhav Chauhan Roll No: 2000301530063*

*Date: 07/02/2024*

**TABLE of CONTENTS**

## CHAPTER NO. TITLE PAGE NO

[ABSTRACT 7](#_bookmark0)

1. [INTRODUCTION](#_bookmark1)
   1. [Problem Definition 8](#_bookmark2)
   2. Background about the project idea 8
   3. Objectives of proposed system 8
   4. Feasibility Study, need and significance. 9
   5. Technical Specification 10
      1. Hardware and Software required 11
2. LITERATURE REVIEW 12
   1. Existing System - Theoretical and methodological

Contributions t o a particular topic of scholarly papers /

articles /books

1. [PROPOSED SYSTEM 16](#_bookmark3)
2. SOFTWARE REQUIREMENT ANALYSIS
   1. [System Architecture 17](#_bookmark4)
   2. Functional Requirements 18

a. Use Case diagram with proper factoring.

* 1. Major Modules and their functionalities 20

[4.4 Software Environment 21](#_bookmark5)

1. SYSTEM ANALYSIS & DESIGN
   1. Activity Diagrams 24
   2. DFDs of the project 25
   3. Input designs 26
   4. Output designs 26
2. IMPLEMENTATION/CORE MODULE
   1. [Used Algorithms/Approaches for projects. 28](#_bookmark6)
   2. [Source Code 31](#_bookmark7)

7. [RESULTS / OUTPUTS & TESTING](#_bookmark8)

* 1. All user interfaces and output screens 39
  2. [Design and Test Steps / Criteria 42](#_bookmark9)

1. CONCLUSIONS 45
2. [REFERENCES 46](#_bookmark10)

# ABSTRACT

Crop Recommendation System for agriculture is based on various input parameters. This proposes a hybrid model for recommending crops to south Indian states by considering various attributes such as Temperature, Rainfall, Humidity, Nitrogen, Potassium and Phosphorus values.

The recommender model is built as a hybrid model using the classifier machine learning algorithm. Based on the appropriate parameters, the system will recommend the crop.

Technology based crop recommendation system for agriculture helps the farmers to increase the crop yield by recommending a suitable crop for their land with the help of geographic and the climatic parameters.

The proposed hybrid recommender model is found to be effective in recommending a suitable crop. Crop yield production value updation has a positive practical significance for guiding agricultural production and for notifying the change in market rate of crop to the farmer.

The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So, the quality of the crops is identified using ranking process. By this process the rate of the low quality and high-quality crop is also notified.

The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of crop which is yielded for further.

## CHAPTER:1 INTRODUCTION

### Problem Definition

Our project aims to develop an advanced Crop Recommendation System leveraging machine learning techniques to optimize crop yield in Indian agriculture. Agriculture, a vital sector in India, faces challenges due to farmers' difficulty in selecting the most suitable crops based on their specific soil requirements. Precision farming is the key approach, and our system seeks to enhance it through precise crop recommendations.

The system will utilize a comprehensive dataset, including soil-specific attributes from a reputed soil testing lab and general crop data. Parameters such as pH, soil colour, Nitrogen, Phosphorous, Potassium and average rainfall will be considered to create a robust foundation for accurate recommendations. The ensemble learning technique, Majority Voting, will be implemented, employing base learners like Support Vector Machine, K- Nearest Neighbors (KNN) and Random Forest.

### Background about Project

Agriculture in India plays a pivotal role in sustaining the economy and providing livelihoods for a significant portion of the population. However, the sector grapples with challenges, especially for small-scale farmers who often lack access to modern technologies and face difficulties in optimizing crop selection based on soil characteristics. The majority of Indian farms are small, with limited irrigation capabilities, contributing to lower productivity levels compared to global standards.

Precision agriculture has emerged as a transformative approach, emphasizing the precise application of inputs like fertilizers and water to maximize crop yields. In this context, the Crop Recommendation System becomes crucial, addressing the prevalent issue of suboptimal crop choices that hinder productivity.

The proposed system builds on the foundation of precision agriculture, utilizing machine learning techniques to provide accurate and efficient crop recommendations. By integrating soil-specific attributes and leveraging ensemble learning methods, the system aims to empower farmers with actionable insights for better decision-making. The initiative aligns with the broader national goal of enhancing agricultural productivity, ensuring food security, and promoting sustainable farming practices. As India seeks to modernize its agricultural sector, the Crop Recommendation System represents a technological leap forward, offering tailored solutions to small and marginal farmers and contributing to the overall growth and resilience of the agriculture industry.

### Objectives

* + - Data set is collected from Kaggle Website. (https://[www.kaggle.com/code/niteshhalai/crop-recommendation-dataset)](http://www.kaggle.com/code/niteshhalai/crop-recommendation-dataset))
    - Data cleaning and pre-processing techniques are applied to convert the raw data into processing data.
    - The data collected is subject to machine learning system along with runtime analysis makes an efficient crop recommendation system.
    - Usage of Ensemble of classifiers makes the model more robust and efficient.
    - Ranking technique used in the project helps us to make efficient decisions.
    - Creating a web application for user registrations and collection of data.
    - The main objective is to obtain a better variety of crops that can be grown over the season. The proposed system would help to minimize the difficulties faced by farmers in choosing a crop and maximize the yield.
    - The model predicts the crop yield by studying factors such as rainfall, temperature, humidity etc.

### Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

###### ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

###### TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

###### SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### Technical Specifications:

###### HARDWARE REQUIREMENTS:

* + - System: Pentium Dual Core.
    - Hard Disk: 40 GB.
    - Monitor: 15’ LED
    - Ram: 4 GB

###### SOFTWARE REQUIREMENTS:

* Operating system: Windows 7/10.
* Coding Language: Python.

## CHAPTER:2 LITERATURE SURVEY

### Existing System

The computational and data demands of structural price forecasting generally far exceed than what is routinely available in developing countries. Consequently, researchers often rely on parsimonious representations of price processes for their forecasting needs. Contemporary parsimonious form of price forecasting relies heavily on time series modelling. In time series modelling, past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. During the past few decades, much effort has been devoted to the development and improvement of time series forecasting models. Time series modelling requires less onerous data input for regular and up-to date price forecasting. Hence there is a need for better classification which would be an ensemble or hybrid classification model.

#### DISADVANTAGES OF EXISTING SYSTEM

* Efficiency is low.
* The existing system which recommends crop yield is either hardware-based being costly to maintain, or not easily accessible.
* Despite many solutions that have been recently proposed, there are still open challenges in creating a user-friendly application with respect to crop recommendation.
* More number of repeated works.

### Title: Crop Prediction using Machine Learning Approaches

##### Authors: Assistant Prof Nischitha K, Dhanush Vishwakarma, Mahendra N,Ashwini, Manjuraju M.R

The paper addresses the challenges faced by farmers in India, where a majority rely on agriculture as their occupation. The repeated cultivation of the same crops without considering soil conditions and the use of same fertilizers lead to reduced crop yields, soil acidification, and top-layer damage. To address these issues, the authors propose a system utilizing machine learning algorithms to help farmers in making informed decisions.

The system's primary objectives are to suggest the most suitable crop for a specific piece of land based on soil content and weather parameters, and to provide information on the required content and quantity of fertilizers, as well as the necessary seeds for cultivation. By incorporating machine learning algorithms such as Support Vector Machine (SVM) for rainfall prediction and Decision Tree for crop prediction, the system aims to enhance farmers' decision-making processes.

### Title: Smart Farming using Machine Learning and Deep Learning Techniques

##### Authors: Senthil Kumar Swami Durai, Mary Divya Shamili

Precise Farming uses advanced technologies such as IOT, Data Mining, Data Analytics, Machine Learning to collect the data, train the systems and predict the results. With the help of technologies Precise farming helps to reduce manual

labor and increase productivity. It guides an individual for smart farming.

This paper uses 10 classification algorithms to find the best model for future prediction. These include Naïve Bayes, Logistic Regression, SVM, Decision Tree, Bagging Classifier, XG Boost Classifier, Random Forest Classifier, Ada Boost Classifier, LGBM

Classifier, Gradient Boosting and KNN. The training and testing ratio was 50:50. Six algorithms obtained a score of more than 90%. The Random Forest model hyper tuned with Randomized CV was selected as the best model since its accuracy is 95.45%.

### Title: Crop Prediction using Machine Learning

##### Authors: Madhuri Shripathi Rao, Arushi Singh, N.V Subba Reddy, Dinesh U Acharya

The paper compares various supervised learning algorithms like KNN, Decision Tree, and Random Forest on the dataset containing 22 varieties of crops. It uses Decision Tree and Random Forest Classifier and evaluates the model's performance under two criterions- Entropy and Gini Index.

It concluded that the crop prediction dataset showed the best accuracy with Random Forest Classifier both in Entropy and Gini Criterion with 99.32%. In contrast, K-Nearest Neighbor with k = 5 has the lowest accuracy among the three with 97.04%, and the accuracy of Decision Tree Classifier is in between KNN and Random Forest Classifier. When comparing the accuracy value, Decision Tree Gini criterion gave a better accuracy of 98.86% compared to Decision Tree Entropy Criterion.

1. **Title: Crop Recommendation System**

##### Authors: Prof. Pradeepa Bandara, Assistant Prof. Thilini Weerasooriya, Ruchirawya T.H., W.J.M. Nanayakkara, Dimantha M.A.C, Pabasara M.G.P

The paper focuses on addressing the challenges faced in Sri Lanka, where despite having manual agricultural knowledge, there is a lack of automated systems to to detect environmental factors and suggest optimal crops for cultivation. It integrates various technologies, including Arduino microcontrollers for environmental data collection, machine learning techniques such as Naïve Bayes and Support Vector Machine. The automatic processing of environmental factors eliminates the need for specialist guidance

and minimizes maintenance costs. With an accuracy exceeding 95%, the system is deemed suitable for Both rural and urban areas in Sri Lanka.

### Title: A Machine Learning Based Crop Recommendation System: A Survey

##### Authors: Rohini Jadhav, Dr. Pawan Bhaladhare

The paper begins by highlighting the crucial role agriculture plays in providing employment and income in rural areas. Despite its importance, the paper notes that the yield per hectare in India is lower than global standards. The authors identify reasons for the high suicide rate among marginal farmers and present their paper as a study offering a solution to address these issues.

The system involves connecting farmers through a smartphone app, utilizing GPS technology for user identification and location. Farmers specify the area and soil type, and machine learning algorithms, including Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), Multivariate Linear Network (MLN), and a combination of regression and KNN, are employed to predict

crop yields. The Random Forest algorithm demonstrated the highest accuracy at 95%. Additionally, the system recommends the use of chemical fertilizers to enhance output.

### Title: Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique

##### Authors: Rohit Kumar Rajak, Ankit Pawar, Mitalee Pendke, Pooja Shinde, Suresh Rathod, Avinash Devare

The paper proposes an ensemble learning approach to enhance prediction accuracy. It uses parameters such as depth, texture, pH, soil color, permeability, drainage, water holding, and erosion. The ensemble technique employed is Majority Voting, with base learners including Support Vector Machine, Naïve Bayes, Multi-layer Perceptron, and Random Forest.

## CHAPTER:3 PROPOSED SYSTEM

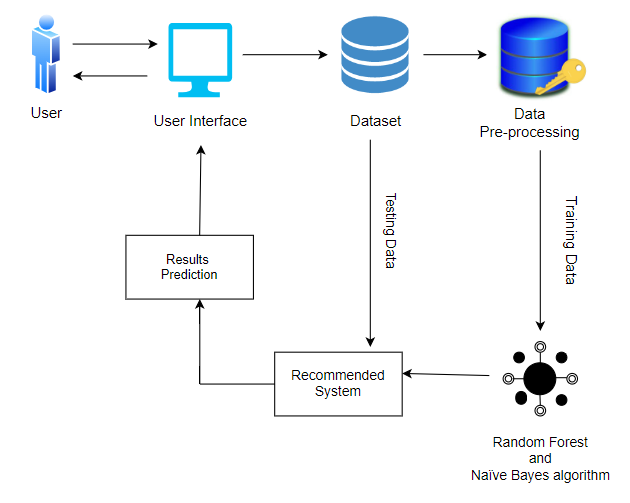
In proposed system, the data analysis technology is used to update the crop yield rate change. The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. The main thing that sets the paper apart from the existing work is that it does not take pH into account for making the predictions. Upon applying data visualization techniques, it was found that the pH parameter has a very less correlation. Removing pH from the training dataset has no effect on the accuracy achieved by including it. Hence, the proposed system does not take the pH value from the user. The proposed work also takes the humidity and the temperature parameter automatically with the help of the user’s current location. It uses the JavaScript geolocation API to first get the user’s latitude and longitude and then uses the Open Weather Map API to find the temperature and humidity based on that. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So, the quality of the crops is identified using ranking process. By this process the rate of the low quality and high-quality crop is also intimated. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This project uses Ensemble of classifiers such as Naïve Bayes and Random Forest classifier. In addition, this project uses Ranking technique.

###### ADVANTAGES OF PROPOSED SYSTEM

* + - Useful to people far away from towns/cities.
    - Better time efficiency.
    - Reduction of repeated work.

## CHAPTER:4 SOFTWARE REQUIREMENTS ANALYSIS

### System Architecture

****

* 1. **USE CASE DIAGRAM**

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of actions that provides something of measurable value to an action and is drawn as a horizontal ellipse.

The Use Case Diagram illustrates the primary interactions between users (Farmers) and the Crop Recommendation System. Here's a detailed description of the key components:

###### Actors:

* + 1. **Farmer:** The primary actor representing users who interact with the Crop

Recommendation System. Farmers play a central role in providing input data and receiving recommendations.

###### Use Cases:

1. **Input Soil Data:**
   * **Description***:* The Farmer can input essential soil-related attributes such as pH, NPK values, humidity, temperature and average rainfall in his area into the system.
   * **Purpose***:* This allows the system to gather crucial information about the soil, forming the basis for accurate crop recommendations.

###### View Recommended Crop:

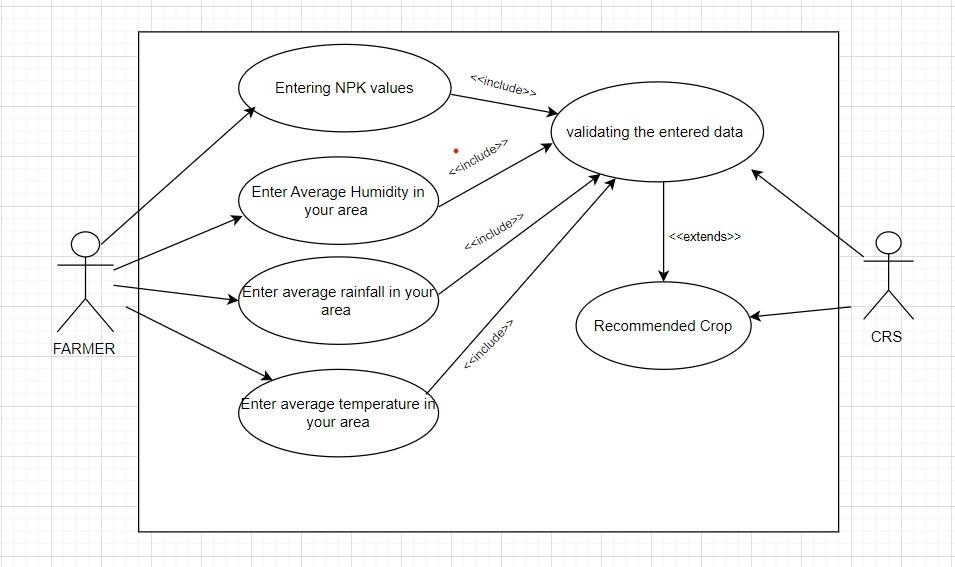
* + **Description***:* The system processes the provided soil data using machine learning models and algorithms to recommend suitable crops.
  + **Purpose***:* Farmers can view the recommended crops tailored to their specific soil conditions, enhancing informed decision-making.

###### Provide Feedback:

* + **Description***:* Farmers have the option to provide feedback on the recommended crops, sharing insights that can contribute to system improvement.
  + **Purpose***:* Continuous feedback helps refine the system's accuracy and ensures it adapts to the dynamic nature of agriculture.

###### Relationships:

* **Uses Relationship (Association):** Farmers (actor) are associated with each use case, indicating their direct involvement in inputting data, receiving recommendations, providing feedback, and accessing historical data.
* **Association between Use Cases and System:** Each use case is connected to the Crop Recommendation System, illustrating the functionalities facilitated by the system.



### Modules used

* Admin Login
* Metadata
* Data Pre-processing
* Crop Recommendation Module

MODULES DESCRIPTION:

#### Admin Login:

This is the first activity, Admin needs to provide a correct contact number and a password, which user enters while registering, in order to login into the webpage. If information provided by the admin matches with the data in the database table, then user successfully login into the webpage else message of login failed is displayed and user need to re-enter correct information.

###### Metadata:

All the main data used in the data set are initialized with the number to use in

the algorithm it is like initializing all the details. In this metadata, we are going to initialize all the crop names with the numbers. This data makes us use the data easily in the algorithm. Hear the metadata of all the crops is given with a particular number. This number is not duplicated that is one number is given to one crop, the same number is not given to the other crop. This metadata consists of more thana hundred crops that grown all over India.

#### Data Pre-processing:

Hear the raw data in the crop data is cleaned and the metadata is appending to it by removing the things which are converted to the integer. So, the data is easy to train. Hear all the data. In this pre-processing, we first load the metadata into this and then this metadata will be attached to the data and replace the converted data with metadata. Then this data will be moved further and remove the unwanted data in the list and it will divide the data into the train and the test data.

#### Crop Recommendation Module:

In this module, we have proposed a model that addresses these issues. The novelty of the proposed system is to guide the farmers to maximize the crop yield by selecting the most profitable crop for the specific region.

### Software Environment

**Python:**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* + - **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
    - **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
    - **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
    - **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### History of Python:

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

#### Python Features:

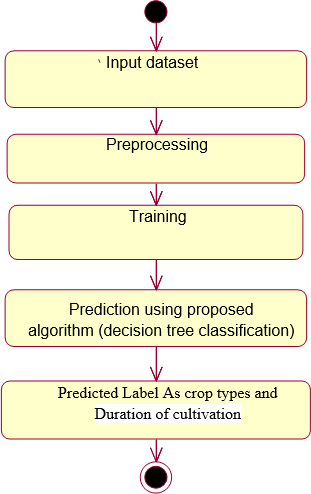
Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

## CHAPTER :5 SYSTEM ANALYSIS AND DESIGN

### ACTIVITY DIAGRAM:

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

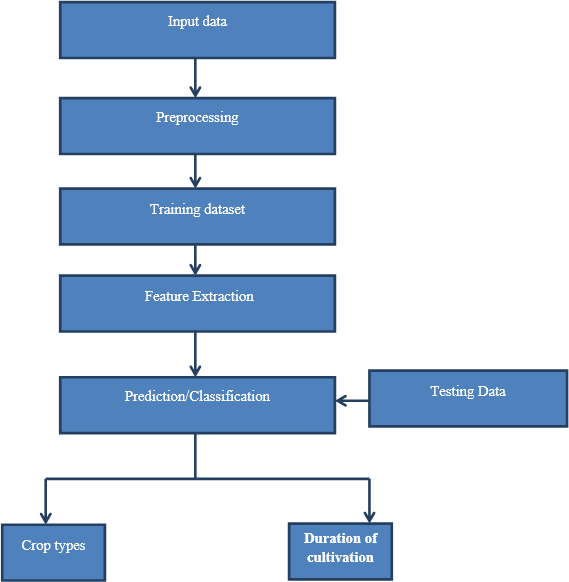


### DATA FLOW DIAGRAM:

* + 1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system,
       1. various processing carried out on this data, and the output data is generated by this system.
    2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the

system and the information flows in the system.

* + 1. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
    2. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



### INPUT DESIGN:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when errors occur.

### OUTPUT DESIGN:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision- making.

* + 1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
    2. Select methods for presenting information.
    3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

## CHAPTER :6 IMPLEMENTATION/CORE MODULE

### Used Algorithms/Approaches for projects.

###### RANDOM FOREST ALGORITHM:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, ***"***Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points. **Step-3:** Choose the number N for decision trees that you want to build. **Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

###### DECISION TREE:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes 18 represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node**.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The complete process can be better understood using the below algorithm:

**Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.

**Step-2:** Find the best attribute in the dataset using Attribute Selection Measure (ASM).

**Step-3:** Divide the S into subsets that contains possible values for the best attributes.

**Step-4:** Generate the decision tree node, which contains the best attribute.

**Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

###### NAIVE BAYES ALGORITHM

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem, which calculates the probability of a hypothesis given the observed evidence. Despite its simplicity, Naive Bayes is widely used for classification tasks, particularly in natural

language processing and spam filtering. The algorithm operates on the assumption of independence between features, hence the "naive" descriptor. This simplifies calculations, making it computationally efficient.

The basic steps of Naive Bayes involve:

* + - 1. **Data Preprocessing:** Gather and prepare the training dataset, ensuring it is representative of the problem. Clean and preprocess the data, handling missing values and irrelevant features.
      2. **Feature Extraction:** Identify relevant features that contribute to the classification task. Calculate the likelihood of each feature given the class labels.
      3. **Training:** Calculate the prior probabilities of each class. Use the Bayes' theorem to compute the posterior probabilities based on the observed features.
      4. **Prediction:** Apply the trained model to new data by calculating the probability of each class given the observed features. Assign the class with the highest probability as the predicted class.

Naive Bayes is efficient for large datasets and is particularly effective when the independence assumption holds true. However, its simplicity may result in suboptimal performance for more complex relationships within the data. Despite this, Naive Bayes remains a popular choice due to its ease of implementation and computational efficiency.

Despite its simplicity and efficiency, Naive Bayes may struggle when faced with highly correlated features or when the independence assumption is significantly violated.

However, its ease of implementation and effectiveness in various contexts have solidified its place in the machine learning landscape as a reliable and efficient algorithm for classification tasks.

###### VOTING CLASSIFIER

A voting classifier in machine learning is an ensemble method that combines multiple individual classifiers to make predictions. This technique leverages the wisdom of the crowd, where each classifier acts as a "voter" and contributes to the final decision. The idea is that by aggregating the opinions of multiple classifiers, the overall performance can be more robust and accurate than that of any individual classifier.

There are two main types of voting classifiers: hard voting and soft voting. In hard voting, each classifier votes for a class, and the class that receives the majority of votes is chosen as the final prediction. Soft voting, on the other hand, involves combining the predicted probabilities of each classifier and selecting the class with the highest average probability.

Voting classifiers are versatile and can be constructed using a variety of base classifiers, such as decision trees, support vector machines, or k-nearest neighbors. They are particularly useful when dealing with diverse and complementary models. This ensemble approach helps mitigate the impact of individual model weaknesses and enhances overall predictive performance. Voting classifiers are employed in various machine learning tasks, including classification and regression, contributing to the robustness and accuracy of predictive models in diverse applications.

### SOURCE CODE

# import libraries import numpy as np import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.preprocessing import LabelEncoder

import warnings warnings.filterwarnings('ignore')

# load dataset and take overview

dataset = pd.read\_csv("../Datasets/Crop\_Recommendation\_Dataset.csv") dataset.head()

# View basic information related to dataset

dataset.info()

# find duplicate rows

series = dataset.duplicated() series[series == True]

# View duplicate rows

dataset.loc[[105,176,274,874,1390]]

# Remove duplicate rows

dataset.drop([105,176,274,874,1390],axis=0,inplace=True)

# count of duplicate rows to cross check that all duplicates has been removed

dataset.duplicated().sum()

# count of null values in every column

col = dataset.columns col = list(col)

for j in col:

print(f"{j} => {dataset[j].isnull().sum()}")

# Drop all rows where any value is null, As number of rows are not much so we can drop them

dataset.dropna(inplace=True) for j in col:

print(f"{j} => {dataset[j].isnull().sum()}")

# View columns present

list(dataset.columns)

# Find crops that are present as label

unique\_crops = dataset['label'].unique() unique\_crops = list(unique\_crops)

unique\_crops

# Find count of datapoints for every crop

dataset['label'].value\_counts()

# plot counts of crop

plt.figure(figsize=(25,8))

sns.countplot(data=dataset,x='label') # effect of Nitrogen on crop

plt.figure(figsize=(15,8)) plt.axes(xlabel="Nitrogen", ylabel="Crop")

sns.scatterplot(data=dataset, x='N', y='label', hue='label') # effect of Phosphorus on crop

plt.figure(figsize=(15,12)) plt.axes(xlabel="Phosphorus", ylabel="Crop") sns.scatterplot(data=dataset, x='P', y='label', hue='label') # effect of Potassium on crop

plt.figure(figsize=(15,12)) plt.axes(xlabel="Potassium", ylabel="Crop")

sns.scatterplot(data=dataset, x='K', y='label', hue='label') # effect of temperature on crop

plt.figure(figsize=(20,12)) plt.axes(xlabel="Temperature", ylabel="Crop")

sns.scatterplot(data=dataset, x='temperature', y='label', hue='label') # effect of humidity on crop

plt.figure(figsize=(22,12)) plt.axes(xlabel="humidity", ylabel="Crop")

sns.scatterplot(data=dataset, x='humidity', y='label', hue='label') # effect of ph on crop

plt.figure(figsize=(20,10)) plt.axes(xlabel="ph", ylabel="Crop")

sns.scatterplot(data=dataset, x='ph', y='label', hue='label') # effect of rainfall on crop

plt.figure(figsize=(20,9)) plt.axes(xlabel="Rainfall", ylabel="Crop")

sns.scatterplot(data=dataset, x='rainfall', y='label', hue='label')

dataset

# crop to digit dict

crop\_to\_digit = { 'rice' : 1,

'maize' : 2,

'chickpea' : 3,

'kidneybeans' : 4,

'pigeonpeas' : 5,

'mothbeans' : 6,

'mungbean' : 7,

'blackgram' : 8,

'lentil' : 9,

'pomegranate' : 10,

'banana' : 11,

'mango' : 12,

'grapes' : 13,

'watermelon' : 14,

'muskmelon' : 15,

'apple' : 16,

'orange' : 17,

'papaya' : 18,

'coconut' : 19,

'cotton' : 20,

'jute' : 21,

'coffee' : 22

}

# digit to crop dict

digit\_to\_crop = { 1 : 'rice',

1. : 'maize',
2. : 'chickpea',
3. : 'kidneybeans',
4. : 'pigeonpeas',
5. : 'mothbeans',
6. : 'mungbean',
7. : 'blackgram',
8. : 'lentil',
9. : 'pomegranate',
10. : 'banana',
11. : 'mango',
12. : 'grapes',
13. : 'watermelon',
14. : 'muskmelon',
15. : 'apple',
16. : 'orange',
17. : 'papaya',
18. : 'coconut',
19. : 'cotton',
20. : 'jute',
21. : 'coffee'

}

# Transform label from string to integer

# Fit and transform the 'Category' column dataset['label'] = dataset['label'].map(crop\_to\_digit)

dataset

# View basic analysis of columns

dataset.describe()

# find correlation matrix

corr\_matrix = dataset.corr() corr\_matrix

# Seperating target variable from features

X = dataset.drop('label',axis=1) Y = dataset['label']

# Splitting Dataset

from sklearn.model\_selection import train\_test\_split

# Assuming 'X' is your feature matrix and 'y' is your target variable

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.15, random\_state=42) print("Types of crop in train data = ", len(Y\_train.value\_counts()))

print("Types of crop in test data = ", len(Y\_test.value\_counts())) from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import GradientBoostingClassifier from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import accuracy\_score from sklearn.metrics import confusion\_matrix

models = {

'Logistic Regression': LogisticRegression(),

'K-Nearest Neighbors': KNeighborsClassifier(), 'Decision Tree': DecisionTreeClassifier(), 'Random Forest': RandomForestClassifier(), 'Naive Bayes': GaussianNB(),

'Support Vector Machine': SVC(), 'Bagging': BaggingClassifier(), 'AdaBoost': AdaBoostClassifier(),

'Gradient Boosting': GradientBoostingClassifier()

}

for model\_name, model in models.items(): # train model

model.fit(X\_train, Y\_train) # predict values for test data

Y\_predict = model.predict(X\_test)

print(f"Accuracy of {model\_name} : {accuracy\_score(Y\_test,Y\_predict)\*100}")

Final Results :

Accuracy of Logistic Regression : 95.74468085106383 Accuracy of K-Nearest Neighbors : 96.96048632218846 Accuracy of Decision Tree : 98.48024316109422 Accuracy of Random Forest : 98.78419452887537 Accuracy of Naive Bayes : 99.3920972644377

Accuracy of Support Vector Machine : 96.96048632218846 Accuracy of Bagging : 99.08814589665653

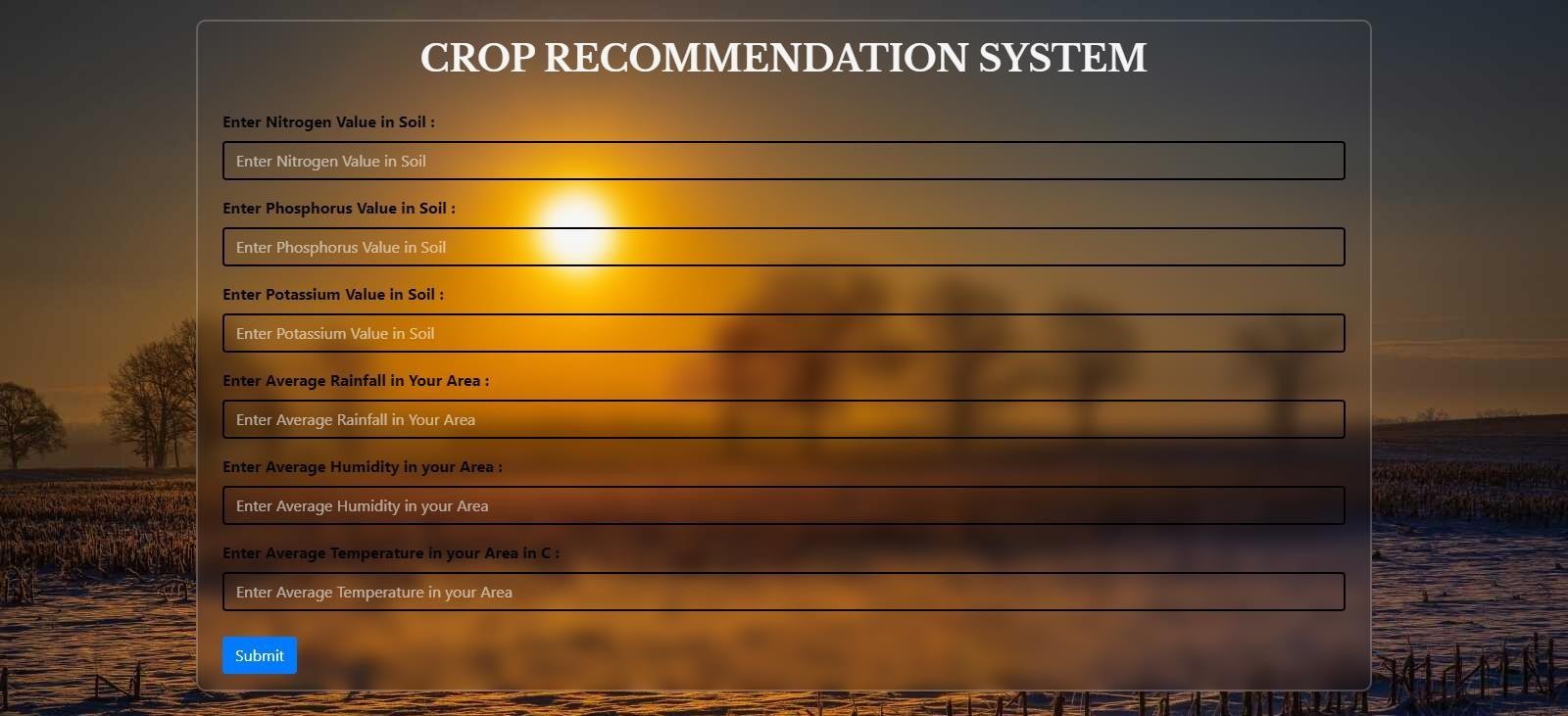
Accuracy of AdaBoost : 17.325227963525837 Accuracy of Gradient Boosting : 97.87234042553192

## CHAPTER : 7 RESULTS / OUTPUTS & TESTING

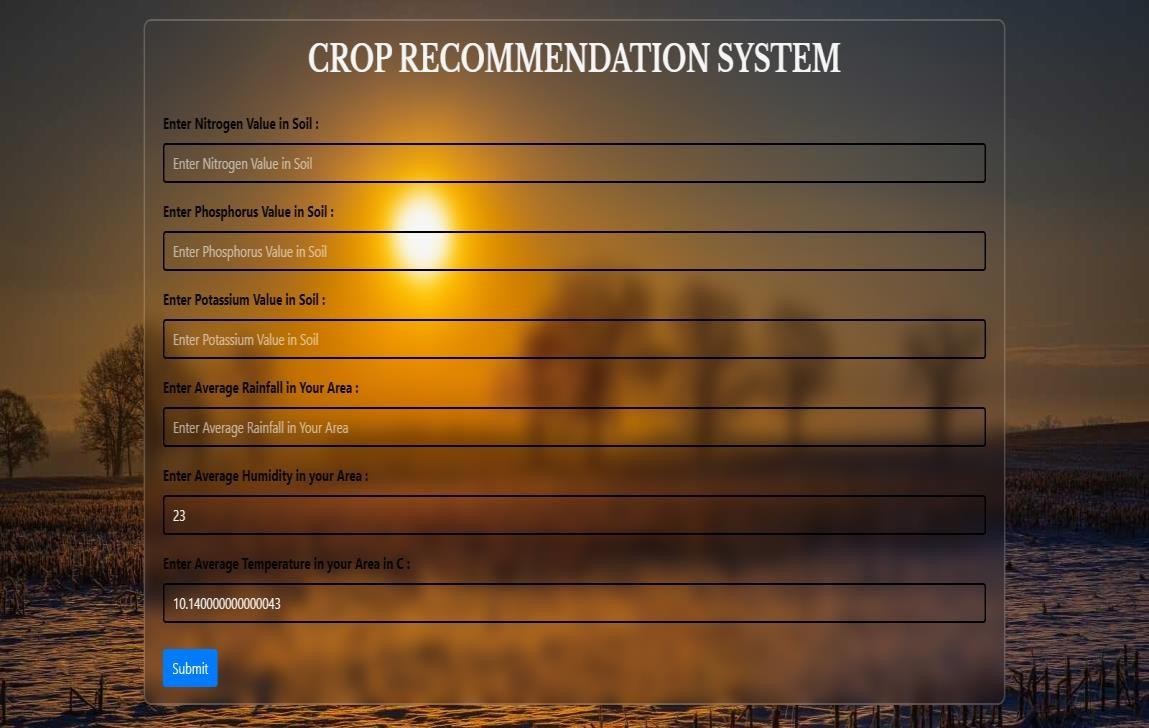
### USER INTERFACE AND ALL SCREEN

In this project, various classifiers are used to obtain the recommendation. Further, a ranking process is applied for decision making in order to select the best classifiers results.

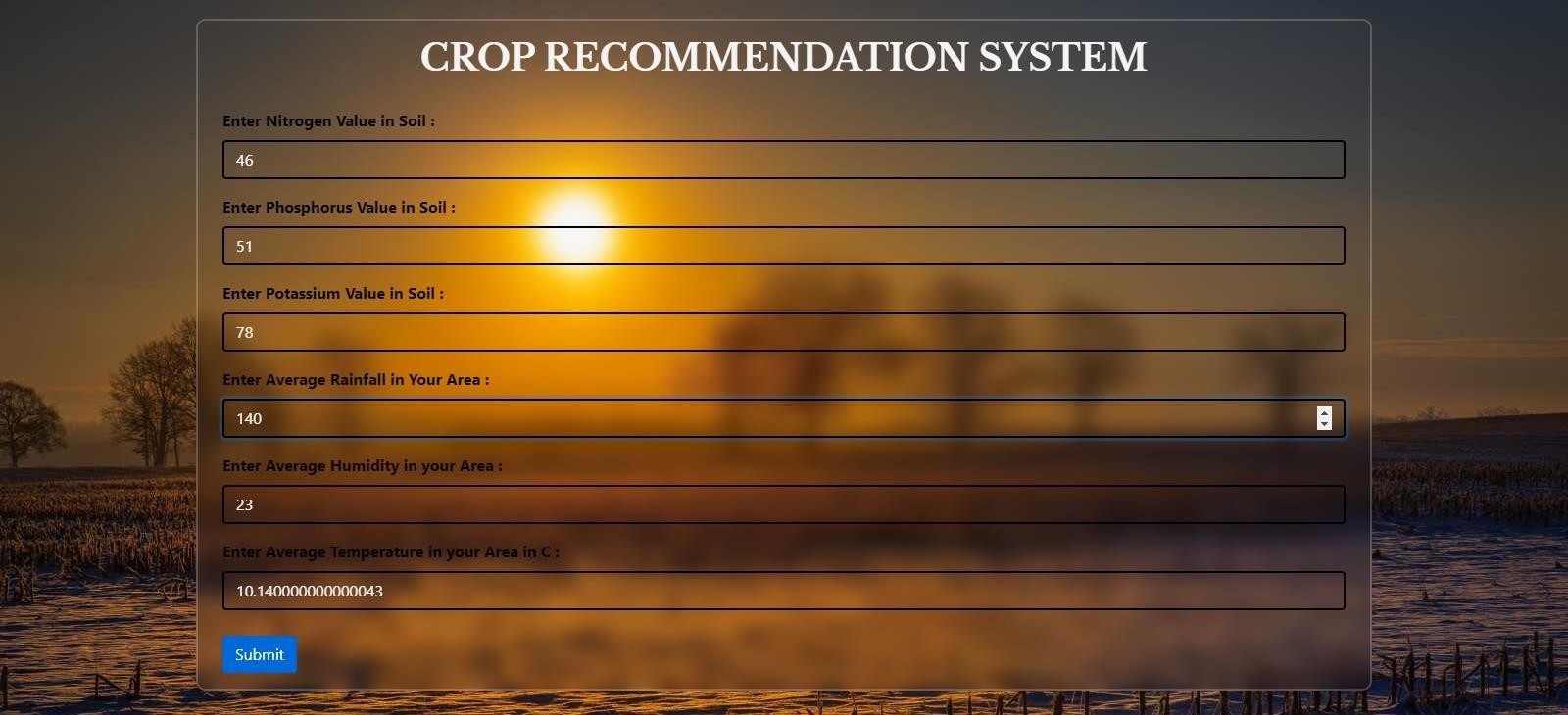
### Screenshot 1:



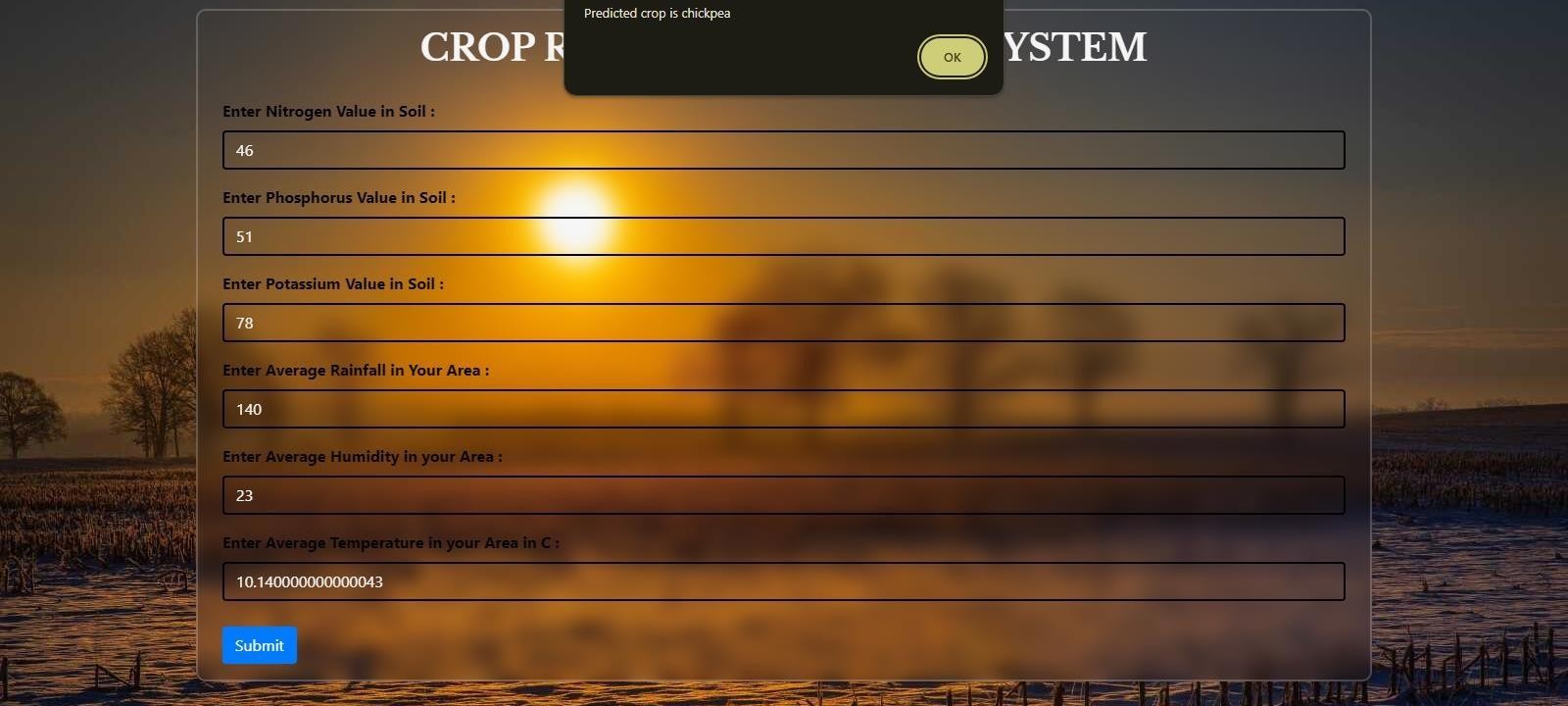
**Screenshot 2:**



### Screenshot 3:



**Screenshot 4:**



## Design and Test Steps / Criteria

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, subassemblies, 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 tests. Each test type addresses a specific testing requirement.

### TYPES OF TESTS:

**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.

### Integration testing:

Integration tests are designed to test integrated software components to determineif 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.

### 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 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. Systems/Procedures: interfacing systems or procedures must be invoked.

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 processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached froma black box level.

### Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box

. you cannot “see” into it. The test provides inputs and responds to outputs without

considering how the software works.

###### Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

###### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

###### Test objectives:

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

###### Features to be tested:

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page

### Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications,

e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER :8 CONCLUSION**

##### This open attitude determines the degree and scope of information sharing. Big data analysis technology can effectively improve the crop yield production as well as profit. This project proposes a novel intelligent system for agricultural crop price prediction by using less parameters. The key idea is to use ensemble of classifiers for prediction. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results.

The solution will benefit farmers to maximize productivity in agriculture, reduce soil degradation in cultivated fields, and reduce fertilizer use in crop production by recommending the right crop by considering various attributes.

This would provide a comprehensive prediction on the basis of geographical, environmental and economic aspects.

**CHAPTER: 9 REFERENCES**

##### Assistant Prof Nischitha K, Dhanush Vishwakarma, Mahendra N, Ashwini, Manjuraju M.R “Crop Prediction using Machine Learning Approaches”

1. Senthil Kumar Swami Durai, Mary Divya Shamili “Smart Farming using Machine Learning and Deep Learning Techniques.”

##### Madhuri Shripathi Rao, Arushi Singh, N.V Subba Reddy, Dinesh U Acharya “Crop Prediction using Machine Learning.”

1. Prof. Pradeepa Bandara, Assistant Prof. Thilini Weerasooriya, Ruchirawya T.H., W.J.M. Nanayakkara, Dimantha M.A.C, Pabasara M.G.P “Crop Recommendation System”

##### Rohini Jadhav, Dr. Pawan Bhaladhare “A Machine Learning Based Crop Recommendation System: A Survey.”

1. Rohit Kumar Rajak, Ankit Pawar, Mitalee Pendke, Pooja Shinde, Suresh Rathod, Avinash Devare “Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique