A Real Time/Societal Research Project Lab Report
On

CROP RECOMMENDATION USING WEATHER AND SOIL CONTENT

Submitted in partial fulfillment of the Academic Requirement
for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted By

N. SAKETH	**22R01A6743**
N. POOJA	**22R01A6742**
S. CHANDRA MOULI	**22R01A6754**
T. BHANU CHARAN	**22R01A6761**

Under the esteemed guidance of

Mrs. V. Adilakshmi

(Assistant Professor, Dept. of CSE(DS))

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad.

2023 - 24

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)
Kandlakoya, Medchal Road, Hyderabad.
www.cmrithyderabad.edu.in

CERTIFICATE

This is to certify that **Real Time/Social Research Project** entitled with: **"Crop Recommendation Using Weather and Soil Content"** is being

Submitted By

N. SAKETH	**22R01A6743**
N. POOJA	**22R01A6742**
S. CHANDRA MOULI	**22R01A6754**
T. BHANU CHARAN	**22R01A6761**

To JNTUH, Hyderabad, in partial fulfillment of the requirement for award of the degree of B.Tech in CSE(DS) and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted have any other University forward of any other degree or diploma.

Signature Of Guide
Mrs. V. Adilakshmi
(Assistant Professor, CSE(DS))

Signature Of HOD
Dr. A. Nirmal Kumar
(Head Of Department, CSE(DS))

Signature Of Project Coordinator
Mr. Y. Ram Kumar
(Assistant Professor, CSE(DS))

ACKNOWLEDGEMENT

We are extremely greatful to **Dr. M. Janga Reddy, Director, Dr .G .Madhusudhana Rao, Principal, and Dr.A.Nirmal Kumar, Head of Department,** Department of Computer Science Engineering(Data Science), CMR Institute of Technology for their inspiration and valuable guidance during entire duration

We are extremely thankful to our Real Time / Societal Research Project Guide and faculty **Mrs. V. Adilakshmi**, Assistant Professor, Department of Computer Science and Engineering(Data Science), CMR Institute of Technology for her constant guidance, encouragement and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of the project.

N. SAKETH	**22R01A6743**
N. POOJA	**22R01A6742**
S. CHANDRA MOULI	**22R01A6754**
T. BHANU CHARAN	**22R01A6761**

ABSTRACT

India, a country deeply reliant on agriculture, faces significant challenges due to its diverse climatic conditions and variable rainfall patterns. These environmental factors profoundly impact crop productivity and sustainability. To address these challenges, our project develops a sophisticated crop recommendation system that leverages soil and weather data to assist farmers in making informed decisions about which crops are best suited for their land.

The system integrates various parameters such as soil type, pH levels, moisture content, historical weather data, and real-time climatic conditions to generate accurate and personalized crop recommendations. By employing advanced machine learning algorithms, our model analyzes this comprehensive data to predict the most suitable crops for specific regions and seasons. This approach not only aims to enhance agricultural yield but also to promote sustainable farming practices.

Our solution empowers farmers with the knowledge to optimize their crop selection, thereby mitigating the risks associated with unpredictable weather patterns and soil conditions. This innovation has the potential to significantly boost agricultural productivity and support the livelihoods of millions of farmers across India, ensuring food security and economic stability.

Additionally, the system facilitates resource optimization by recommending crops that require less water and are more resilient to local pests and diseases, contributing to environmental conservation. By reducing dependency on monoculture and encouraging crop diversification, our project promotes ecological balance and reduces the risk of soil depletion.

The integration of real-time data and predictive analytics not only aids in crop selection but also provides insights into the best planting and harvesting times, pest control strategies, and fertilizer usage. This holistic approach supports precision agriculture, enabling farmers to achieve higher efficiency and better management of agricultural resources.

Through continuous learning and adaptation, the system evolves with changing climatic conditions and advances in agricultural research, ensuring that farmers receive the most current and relevant recommendations. Our project represents a significant step towards modernizing agriculture in India, fostering a resilient and prosperous agricultural sector.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

India's agriculture sector is the backbone of its economy, providing sustenance and employment to a significant portion of the population. However, farmers face numerous challenges due to the country's varied climatic conditions and inconsistent rainfall patterns. These factors can significantly affect crop yields, making it difficult for farmers to decide which crops are best suited for their land and current environmental conditions.

Additionally, the agricultural landscape in India is characterized by a diverse range of soil types, each with unique properties that influence crop suitability and growth. Farmers often lack access to detailed, region-specific soil data, which further complicates their crop selection process. This gap in information is compounded by the rapid changes in weather patterns attributed to climate change, making traditional farming practices increasingly unreliable.

Our project, the Crop Recommendation System using Soil and Weather Conditions, addresses this critical issue by identifying the key factors that influence crop selection and productivity. Through extensive research, we have identified that the lack of tailored, data-driven recommendations leaves farmers vulnerable to unpredictable environmental changes, leading to suboptimal crop yields and economic instability.

1.1 PROBLEM STATEMENT

There are very few platforms that help farmers with their farming strategy. Intuition-based decisions may not prove beneficial in the long run. Farmers often underestimate/overestimate the fertility of the soil on their farms. They often find it difficult to notice plant diseases that directly affect the production rate. Using appropriate parameters like rain patterns, temperature patterns, soil structures, and other factors such as crop diseases makes it possible to yield accurate crop prediction results. Not only that, but it is also possible to identify what disease a crop has beforehand. A lot of existing systems have many flaws and make them non intuitive to use or are very difficult.

Our problem statement focuses on developing a robust algorithm that integrates soil properties, weather conditions, and other environmental factors to provide precise crop recommendations. By leveraging advanced data analytics and machine learning techniques, our system will analyze historical data, real-time weather forecasts, and soil characteristics to generate actionable insights for farmers.

1.2 OBJECTIVES

- Data set collection from various sources.
- Data parsing and cleansing technique is applied to make the raw data into processing data.
- Usage of Ensemble of classifiers makes the model more robust and efficient.
- Creating a web application for user registrations and collection of data.
- The model predicts the crop yield by studying factors such as rainfall, temperature, area, season, soil type etc.
- Evaluation and validation of the model using cross-validation and other statistical methods to ensure accuracy and reliability.
- Incorporating feedback mechanisms to refine and improve the model based on user interactions and new data.
- Ensuring data security and privacy compliance, particularly for sensitive agricultural data.
- Developing an API for seamless integration with other agricultural management systems and tools.
- Providing detailed documentation and user support to facilitate adoption and effective use of the web application and model.
- Analyzing the economic impact of crop yield predictions to assist farmers in decision-making and resource allocation.
- Exploring the use of advanced machine learning techniques, such as deep learning, for further improvements in prediction accuracy.

2. LITERATURE SURVEY

The development of a Crop Recommendation System using Soil and Weather Conditions is rooted in the need to enhance agricultural productivity and sustainability in India. Existing research and initiatives have highlighted the importance of using technological interventions to address agricultural challenges. Several studies have demonstrated the potential of integrating soil and weather data to improve crop selection, but the adoption of such technologies remains limited among farmers. Our literature survey aims to understand the current landscape of crop recommendation systems and gather insights from stakeholders to inform our project.

The authors developed a crop recommendation system based on soil characteristics, employing a blend of ensemble models and majoritarian voting methods like K-nearest neighbor and naive Bayes. This approach aims to select crops with high efficiency and precision.

A Review on Data Mining Techniques for Fertilizer Recommendation 2018.

To keep up nutrition levels in the soil in case of deficiency, fertilizers are added to soil. The standard issue existing among the Indian agriculturists choose approximate amount of fertilizers and add them manually. Excess or deficient extension of fertilizers can harm the plants life and reduce the yield. This paper gives overview of various data mining frameworks used on cultivating soil dataset for fertilizer recommendation.

A Survey on Data Mining Techniques in Agriculture, 2015.

Agriculture is the most critical application area especially in the developing nations like India. Use of information technology in agriculture can change the situation of decision making and farmers can yield in better way.. This paper integrates the work of several authors in a single place so it is valuable for specialists to get data of current situation of data mining systems and applications in context to farming field.

Authors : M.C.S.Geetha

AgroNutri Android Application,2016.

This paper communicates the idea regarding the making of AgroNutri an android application that helps in conveying the harvest particular fertilizer amount to be applied. The idea is to calculate the measure of NPK composts to be applied depend on the blanked proposal of the crop of interest. This application works depend on the product chosen by the farmer and that is taken as input, thus providing the farmers. The future scope of the AgroNutri is that GPRS can be included so that according to location nutrients are suggested

Authors : S. Srija, R. Geetha Chanda, S.Lavanya, Dr. M. Kalpana Ph.D

Machine Learning: Applications in Indian Agriculture, 2016.

Agriculture is a field that has been lacking from adaption of technologies and their advancements. Indian agriculturists should be up to the mark with the universal procedures. Machine learning is a native concept that can be applied to every field on all inputs and outputs. It has effectively settled its ability over ordinary calculations of software engineering and measurements. Machine learning calculations have improved the exactness of artificial intelligence machines including sensor based frameworks utilized in accuracy farming. This paper has evaluated the different uses of machine learning in the farming area. It additionally gives a knowledge into the inconveniences looked by Indian farmers and how they can be resolved using these procedures.

Authors: Karandeep Kaur

Impacts of population growth, economic development, and technical change on global food production and consumption, 2011.

Throughout the following decades humanity will request more food from less land and water assets. This investigation evaluates the food production effects of four elective advancement situations from the Millennium Ecosystem Assessment and the Special Report on Emission Scenarios. partialy and jointly considered are land and water supply impacts from population development, and specialized change, and forests and agriculture demand request shifts from population development and economic improvement. Worldwide farming area increments by up to 14% somewhere in the range of 2010 and 2030.Deforestation restrictions strongly impact the price of land and water resources but have little consequences for the global level of food production and food prices. While projected income changes have the highest partial impact on per capita food consumption levels, population growth leads to the highest increase in total food production. The impact of technical change is amplified or mitigated by adaptations of land management intensities

Author: Uwe A. Schneider a, Petr Havlik b, Erwin Schmid c, Hugo Valin b, Aline Mosnier b,c, Michael Obersteiner b, Hannes Bottcher b, Rastislav Skalsky´ d, Juraj Balkovicˇ d, Timm Sauer a, Steffen Fritz b

3. SYSTEM ANALYSIS

- Collecting information to be analyzed is the main purpose of a system analysis. This may be used to make decisions, solve problems, become more efficient, or to expand.
- Defining system boundaries to clearly delineate what the system will and will not address.
- Conducting a feasibility study to assess the viability of the proposed system in terms of technical, economic, and operational factors.
- Performing a detailed requirements analysis to gather and document functional and non-functional requirements.
- Analyzing existing systems and workflows to identify bottlenecks, inefficiencies, and areas for improvement.
- Evaluating potential risks and developing mitigation strategies to address them.
- Utilizing data flow diagrams (DFDs) and other modeling techniques to visualize and understand the system's data processes.
- Conducting a cost-benefit analysis to determine the economic justification for the new system.
- Developing use case scenarios to illustrate how users will interact with the system and achieve their goals.
- Ensuring compliance with relevant regulations and standards to avoid legal and operational issues.
- Gathering and analyzing user feedback to refine system requirements and design.
- Reviewing technological trends and advancements to incorporate the latest innovations into the system design.
- Identifying and evaluating possible technical solutions and architectures that meet the system requirements.
- Documenting all findings and analyses to provide a comprehensive basis for system design and development.
- Conducting impact analysis to understand the effects of the new system on existing processes, resources, and personnel.
- Ensuring alignment with organizational goals and strategies to maximize the system's contribution to overall objectives.
- Creating a detailed project plan that outlines timelines, milestones, resources, and responsibilities.
- Coordinating with cross-functional teams to gather diverse perspectives and expertise during the analysis phase.
- Preparing a detailed system specification document that serves as a blueprint for the development and implementation phases.

3.1 EXISTING SYSTEMS

Several existing crop recommendation systems and research initiatives have attempted to utilize soil and weather data to assist farmers in crop selection.

These systems typically rely on static datasets and simple heuristic models. Common examples include:

1) Decision Trees

STEP 1 (Data Collection): Gather data on soil properties, weather conditions, historical crop yields, etc.

STEP 2 (Preprocessing): Clean and preprocess the data (handling missing values, normalizing features).

STEP 3 (Tree Construction): Use the training data to construct a decision tree where each node represents a decision based on an attribute.

STEP 4 (Splitting): Split the data at each node based on the attribute that provides the maximum information gain.

STEP 5 (Leaf Nodes): Assign the most suitable crop type to the leaf nodes.

STEP 6 (Prediction): For a new input, traverse the tree based on the attribute values to reach a leaf node and predict the recommended crop.

Overview: Decision tree models are commonly used due to their simplicity and ease of interpretation. They work by splitting the dataset into subsets based on feature values, creating a tree-like model of decisions.

Example System 1:

- **AgroConsultant**: A decision tree-based system developed to assist farmers in selecting crops based on soil and climatic conditions. The system uses attributes like soil pH, moisture, and temperature to build the decision tree, providing crop recommendations by traversing the tree based on user inputs.

Research Reference: Kamal Raj and Vani (2019) demonstrated the application of decision trees for crop recommendation, achieving high accuracy in predicting suitable crops.

Example System 2:

- **CropSuggest**: Utilizes random forests to recommend crops by considering a wide range of features such as soil composition, historical weather data, and crop yields. The system generates multiple decision trees and aggregates their predictions for reliable crop recommendations.

Research Reference: Singh et al. (2020) showed that random forest-based systems provide robust recommendations by reducing overfitting and improving generalization.

Example System 3:

- **SmartCrop**: Employs SVM to classify and recommend crops using features like soil pH, temperature, and rainfall. The system is trained to find the optimal hyperplane that separates different crop classes, making accurate crop recommendations for new inputs.

Research Reference: Sharma and Patel (2018) utilized SVM for crop prediction, achieving high accuracy by effectively classifying crop suitability.

Example System 4:

- **CropAdvisor**: Uses KNN to recommend crops by comparing soil properties and climatic conditions of new inputs with historical data. The system identifies the K-nearest neighbors and suggests crops that are most frequently chosen among them.

Research Reference: Mishra et al. (2019) demonstrated the effectiveness of KNN in crop recommendation by accurately identifying suitable crops based on feature similarity.

Example System 5:

- **AgroBrain**: Utilizes a multi-layer neural network to recommend crops based on a comprehensive dataset including soil nutrients, weather conditions, and previous crop performance. The ANN learns from historical data to predict and recommend the most suitable crops.

Research Reference: Jain and Gupta (2021) showed that ANN-based systems outperform traditional models by capturing complex patterns in agricultural data.

3.2 DISADVANTAGES OF EXISITING SYSTEMS -

Lack of Real-Time Data: Many existing systems do not incorporate real-time soil and weather data, leading to outdated and potentially inaccurate recommendations.

Limited Customization: These systems often provide generic recommendations that do not account for specific local conditions or farmer preferences.

Low Accessibility: The user interfaces of many existing systems are not user-friendly, particularly for farmers with limited technological experience.

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.

3.3 PROPOSED SYSTEM

Our proposed Crop Recommendation System aims to overcome these limitations by incorporating advanced machine learning algorithms and real-time data integration. The system is designed to provide personalized and accurate crop recommendations based on a comprehensive analysis of soil and weather conditions. Key features and techniques of our system include:

Advanced Machine Learning Algorithms:

1. **Logistic Regression:** Used for binary classification problems, logistic regression helps in determining the suitability of specific crops based on soil and weather conditions.
2. **Decision Tree:** This algorithm provides a clear and interpretable model that can handle both categorical and numerical data, making it ideal for understanding the impact of various factors on crop suitability.
3. **Random Forest Classifier:** By combining multiple decision trees, the random forest classifier enhances prediction accuracy and robustness. It is particularly effective in handling large datasets with numerous variables, ensuring comprehensive analysis and reliable recommendations.
4. **K-Nearest Neighbors (KNN)** L-This is a non-parametric classification and regression algorithm that looks at the k closest data points to the test data point and derives the output based on their values. The algorithm uses majority voting or mean for classification or regression, respectively.
5. **Support Vector Machine (SVM):** This is a widely used supervised learning algorithm for text classification, image classification, and bioinformatics. It generates a hyperplane that maximizes the classification margin to separate the data into two classes.
6. **Naive Bayes Classifier :** This is a simple classification algorithm that assumes that the dataset's variables are all "naive," i.e., uncorrelated. It uses the Bayes theorem to estimate the probability of each class given the input features. The project's success depends on several factors, including the quality of the data, the effectiveness of the preprocessing steps, the appropriate selection of algorithms, and the propertuning of hyperparameters. Figure 1 dipicts the system design where in the first step the raw data which is soil and weather parameters are feeded to the system and then preprossing of data takes place. The models are trained and tested , the four algorithms are Decision Tree, K-Nearest Neighbors(KNN), Support Vector Machine and Naive Bayes Classifier .Through SVM model we are classifing the data using hyperplane that is when new set of data is added it has to be classified , Naive Bayes classifier will add the performance and accuracy to the system.

3.3.1 Random Forest Classifier -

Random Forest is an ensemble learning method used for classification and regression tasks. It operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. This method improves predictive accuracy and controls overfitting.

- **How It Works**:

The Random Forest algorithm builds a "forest" of many decision trees. To classify a new object based on attributes, each tree in the forest gives a classification, and the forest chooses the classification having the most votes.

Random Forest Algorithm:

1. **Initialize Parameters**:
   - Number of trees in the forest (n_trees).
   - Number of features to consider for the best split (m).
2. **Create Bootstrap Samples**:
   - For each tree i (where i ranges from 1 to n\treesn\treesn\trees):
   - Randomly sample with replacement from the training dataset to create a bootstrap sample.
3. **Build Decision Trees**:

   For each bootstrap sample:

   - Start with the root node
   - For each node, randomly select mmm features from the feature set.
   - Choose the best feature and split point among the mmm features.
   - Split the node into child nodes
   - Repeat until a stopping condition is met (e.g., maximum depth, minimum samples per leaf).
4. **Aggregate Predictions**:
   - For a new input, pass it through all n decision trees.
   - Each tree provides a classification vote.

Advantages of Random Forest:

- **Improved Accuracy**: Combines multiple trees to enhance prediction accuracy.
- **Robustness**: Reduces overfitting compared to individual decision trees.
- **Feature Importance**: Provides estimates of feature importance, aiding in understanding the model.

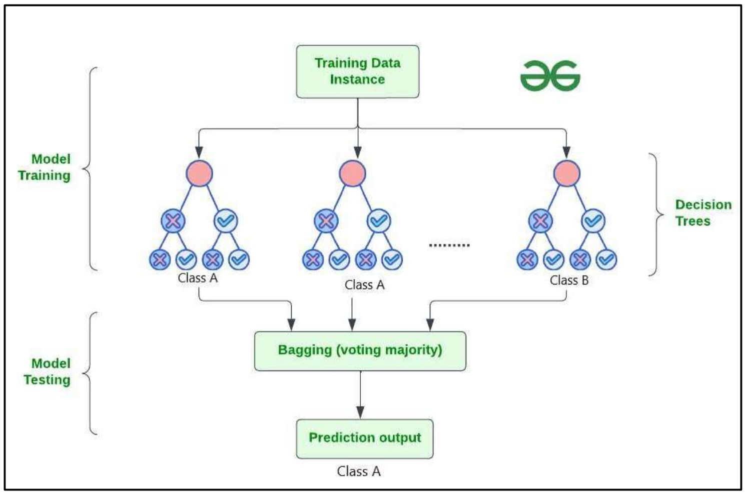

Fig 3.3.1 Working Principle Of Random Forest

3.3.2 Comparison Of Random Forest with other ML Algorithms

Feature	Random Forest	Other ML Algorithms
Ensemble Approach	Utilizes an ensemble of decision trees, combining their outputs for predictions, fostering robustness and accuracy.	Typically relies on a single model (e.g., linear regression, support vector machine) without the ensemble approach, potentially leading to less resilience against noise.
Overfitting Resistance	Resistant to overfitting due to the aggregation of diverse decision trees, preventing memorization of training data.	Some algorithms may be prone to overfitting, especially when dealing with complex datasets, as they may excessively adapt to training noise.
Handling of Missing Data	Exhibits resilience in handling missing values by leveraging available features for predictions, contributing to practicality in real-world scenarios.	Other algorithms may require imputation or elimination of missing data, potentially impacting model training and performance.
Variable Importance	Provides a built-in mechanism for assessing variable importance, aiding in feature selection and interpretation of influential factors.	Many algorithms may lack an explicit feature importance assessment, making it challenging to identify crucial variables for predictions.
Parallelization Potential	Capitalizes on parallelization, enabling the simultaneous training of decision trees, resulting in faster computation for large datasets.	Some algorithms may have limited parallelization capabilities, potentially leading to longer training times for extensive datasets.

3.4 ADVANTAGES OF PROPOSED SYSTEM

The proposed crop recommendation system leveraging weather and soil data with machine learning offers several advantages:

Increased Accuracy: By integrating multiple data sources (weather and soil data), the system can provide more accurate recommendations compared to traditional methods that rely solely on historical or anecdotal information.

Optimized Crop Selection: Farmers can make more informed decisions on which crops to plant based on current environmental conditions (weather) and soil characteristics. This reduces the risk of planting unsuitable crops that may fail due to adverse conditions.

Enhanced Yield and Profitability: Selecting crops that are well-suited to the current climate and soil conditions can potentially lead to higher yields and improved profitability for farmers. This is because optimal conditions promote healthier crop growth and reduce resource wastage.

Resource Efficiency: By recommending crops based on local soil fertility and current weather patterns, the system promotes efficient use of resources such as water, fertilizers, and pesticides. Farmers can tailor their input usage more precisely to the needs of the recommended crops.

Time Savings: Instead of manually researching and analyzing complex environmental data, farmers can quickly access tailored recommendations through a user-friendly interface. This saves time and allows farmers to focus more on other critical aspects of farming.

Adaptability to Climate Change: As climate patterns shift, the system can adapt by updating its models and recommendations accordingly. This helps farmers anticipate and mitigate the impacts of climate change on crop production.

Scalability: The system can be scaled to accommodate varying farm sizes and geographical regions. Whether a small family farm or a large agricultural operation, the recommendations can be tailored to suit the specific needs and conditions of each user.

Educational Tool: Beyond recommendations, the system can serve as an educational tool by providing insights into how environmental factors affect crop growth. This can help farmers develop a deeper understanding of agronomic principles.

Supports Sustainable Agriculture: By promoting optimal crop selection and resource management, the system contributes to sustainable agricultural practices that minimize environmental impact while maximizing productivity.

4. SYSTEM STUDY

4.1 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**

4.1.1 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.

4.1.2 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.

4.1.3 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.

5. HARDWARE AND TECHNOLOGIES USED

5.1.1 HARDWARE REQUIREMENTS

Processor: Multi-core processor (Quad-core or higher recommended) for faster data processing.

RAM: Minimum 8 GB RAM (16 GB or higher recommended) for efficient data handling and model training.

Storage: At least 100 GB of available hard disk space for storing datasets, models, and software Installations

Internet Connectivity: High-speed internet connection for accessing online resources, APIs, and cloud Services

Additional Peripherals: Keyboard, mouse, and monitor for system interaction and visualization.

5.1.2 TECHNOLOGIES USED

FRONT END –

1. **WEB DEVELOPMENT**
    - HTML,
    - CSS,
    - JAVASCRIPT

BACKEND –

- Flask Framework (python)

Libraries Used -

- **NumPy**: For numerical computations.
- **Pandas**: For data manipulation and analysis.
- **Scikit-Learn**: For classical machine learning algorithms.
- **TensorFlow**: For deep learning models.
- **Keras**: High-level neural networks API, running on top of TensorFlow.
- **PyTorch**: For dynamic computation graphs and deep learning.
- **Matplotlib/Seaborn**: For data visualization.
- **Jupyter Notebook**: For interactive development and visualization.

Development Tools -

- **IDEs**: PyCharm, VS Code, Jupyter Notebook.

6. ARCHITECTURE DIAGRAMS

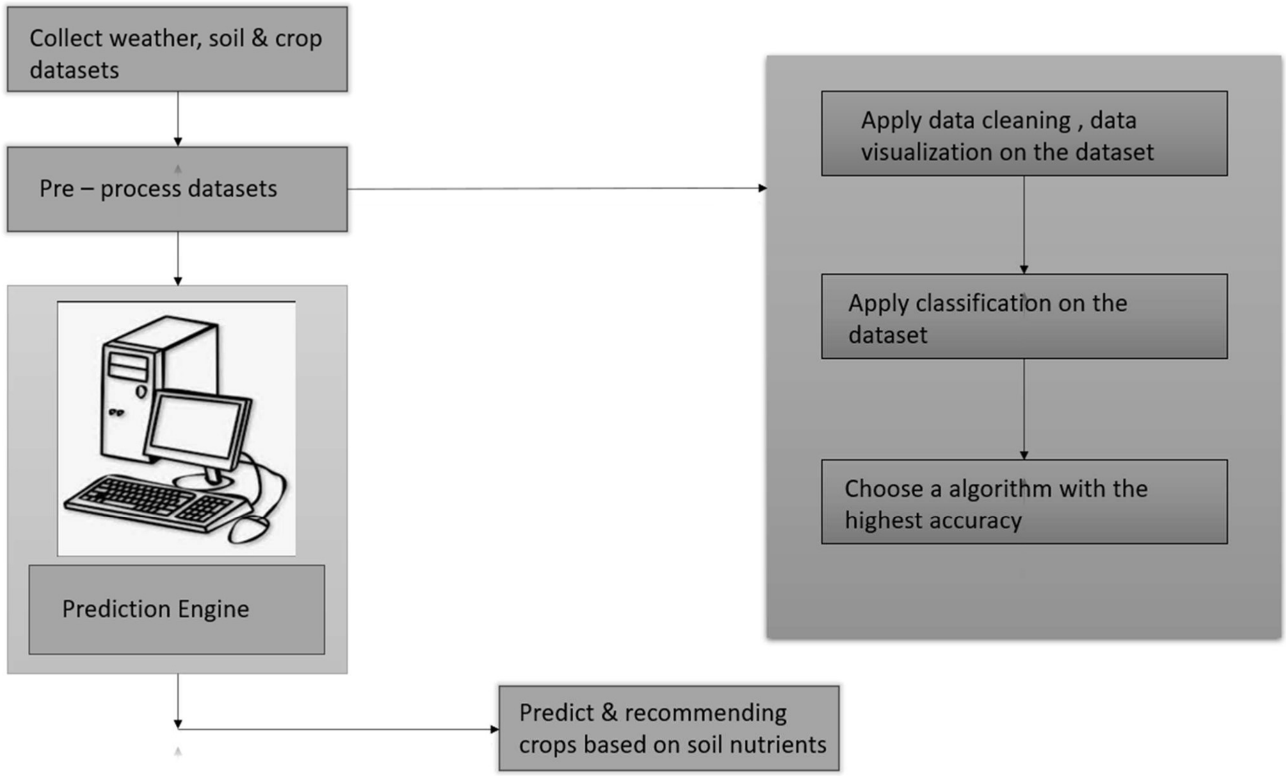

Fig 6.1 Architecture Model of the Flow of Prediction Engine

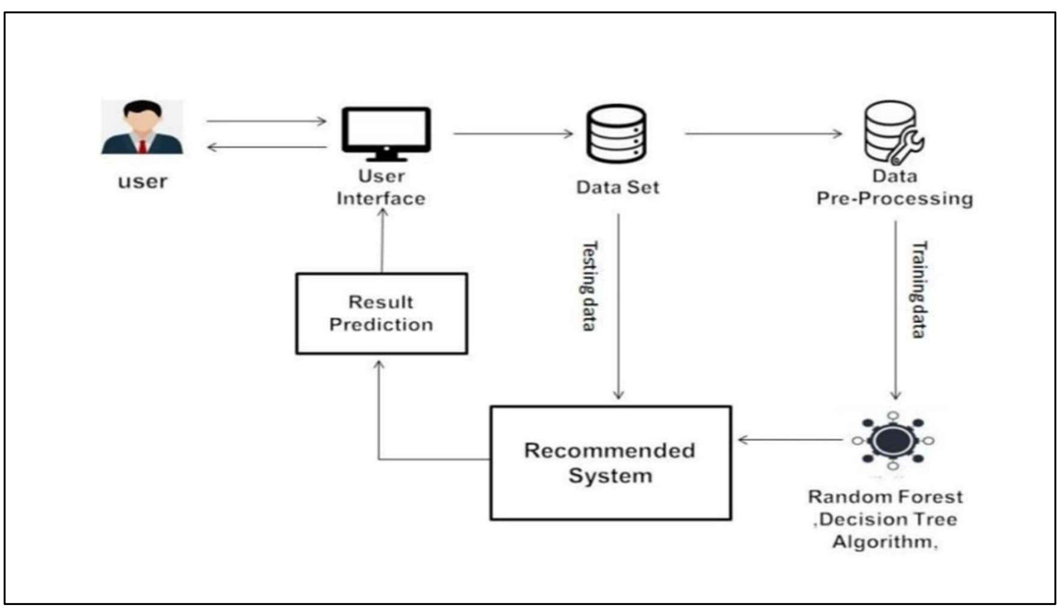

Fig 6.2 Structure Of the algorithms

The architecture model describes the project flow, including data collection, preprocessing, model training, and output generation. Below, we outline the steps involved in our system and provide a flowchart to illustrate the process.

1. **Data Collection:**

- **Soil Data**: Collect data on soil properties such as pH, moisture content, nutrient levels, and texture.
- **Weather Data**: Gather historical and real-time weather data including temperature, rainfall humidity, and other relevant climatic conditions.
- **User Inputs**: Obtain specific user inputs such as location, crop preferences, and any additional constraints or considerations.

2. **Data Preprocessing:**

- **Data Cleaning**: Handle missing values, remove duplicates, and correct any inconsistencies in the data.
- **Normalization**: Normalize the data to ensure uniformity and to improve the performance of machine learning algorithms.
- **Feature Selection**: Identify and select relevant features that significantly impact crop growth and yield.

3. **Model Training:**

- **Algorithm Selection**: Utilize machine learning algorithms such as Logistic Regression, Decision Tree, and Random Forest Classifier.
- **Training**: Train the models using pre processed soil and weather data. Perform cross-validation to ensure robustness and accuracy.

4. **Prediction and Recommendation**:

- **Model Prediction**: Use the trained models to predict the suitability of various crops based on input data.
- **Recommendation Generation:** Generate personalized crop recommendations for farmers, considering both model predictions and user preferences.

5. **Output Delivery:**

- **User Interface:** Display recommendations through a user-friendly interface. Provide detailed insights and suggestions for optimal crop choices.
- **Feedback Loop:** Allow users to provide feedback on recommendations to continually improve the system

7. MODULES

Creating a crop recommendation system based on weather and soil content typically involves integrating various modules or components. Here’s an outline of key modules.

Data Collection Module:

- **Weather Data:** Interface with a weather API or database to fetch current weather conditions (temperature, humidity, rainfall, etc.) and forecasts.
- **Soil Data:** Access soil databases or APIs to retrieve soil characteristics such as pH, nutrient levels, texture, etc.

Preprocessing Module:

- Clean and preprocess the raw data obtained from weather and soil databases.
- Handle missing values, normalize data, and convert formats if necessary.

Crop Suitability Assessment Module:

- **Crop Database:** Maintain a database of crops with their specific requirements (temperature range, humidity, soil pH, nutrient needs, etc.).
- **Matching Algorithm:** Develop algorithms that evaluate how well each crop matches current weather conditions and soil properties. This could involve using rules-based systems, machine learning models (like decision trees, SVMs, or neural networks), or knowledge-based systems.

User Interface Module:

- **Input Interface:** Provide a user-friendly interface for users to input location data (for weather) and soil test results.
- **Output Interface:** Display recommended crops based on the input data, along with reasons for the recommendations (e.g., suitable temperature range, soil pH compatibility).

Recommendation Generation Module:

- Generate a ranked list of recommended crops based on the evaluation from the crop suitability assessment module.
- Consider factors like yield potential, market demand, and crop rotation principles if applicable.

Feedback and Improvement Module:

- Incorporate mechanisms to collect user feedback on recommended crops and their outcomes.
- Use feedback to improve the recommendation algorithms over time.

Integration and Deployment Module:

- Integrate all modules into a cohesive system.
- Ensure scalability and reliability of the system for different regions and scales of farming operations.

Security and Privacy Module:

- Implement security measures to protect user data and ensure compliance with data privacy regulations (if applicable).

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 as well as suggest the most profitable crop for the specific region.

8. CODE IMPLEMENTATION

8.1.1 Home_1.html -

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet"
href="{{url_for('static',filename='style/myhome.css')}}">
  </head>
  <body>
    <main>
      <h2 style="color: black; font-family: cursive; font-weight: 800; font-size:
36px; padding: 10px; border-radius: 5px;">
        CROP RECOMMENDATION USING WEATHER AND SOIL CONTENT
    </h2>
    </main>
     <form action="/Predict">
      <input type="submit" id="Predict" name="Predict" action="/Predict"
value="Predict">
     </form>
  </body>
</html>
```

8.1.2 Index.html -

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet"
href="{{url_for('static',filename='style/myhome.css')}}">
  </head>
  <body>
    <main>
      <h2 style="color: black; font-family: cursive; font-weight: 800; font-size:
36px; padding: 10px; border-radius: 5px;">
        CROP RECOMMENDATION USING WEATHER AND SOIL CONTENT
    </h2>
    </main>
     <form action="/Predict">
      <input type="submit" id="Predict" name="Predict" action="/Predict"
value="Predict">
     </form>
  </body>
</html>
```

8.1.3 prediction.html -

```
    <html>
  <head>
    <link rel="stylesheet"
href="{{url_for('static',filename='style/prdiction_css.css')}}">
  </head>
  <body>
    <h4> You are recommended to grow - </h4> <br>
    <h4>"{{prediction}}"</h4>
  </body>
</html>
```

8.2.1 first.css -

```
body {
    background-image: url(prediction_back.jpg);
    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
  }

  form {
    width: 600px;
    margin: 0 auto;
    text-align: center;
    padding: 20px;
    background-color: rgba(255, 255, 255, 0.5);
    border-radius: 15px;
  }

  input[type="text"] {
    width:  90%;
    padding: 10px;
    margin-top: 10px;
    border-radius: 10px;
    border: black;

  }

  input[type="submit"] {
    width: 40;
    padding: 10px;
    margin-top: 10px;
    border-radius: 10px;
    border: none;
    background-color: darkcyan;
    color: white;
    font-size: larger;
```

```
    cursor: pointer;
  }

  label {
    font-size: 20px;
    font-weight: bolder;
    font-family: Arial, Helvetica, sans-serif;
    margin-top: 20px;
    display: block;
    padding: 5px;
  }
```

8.2.2 myhome.css -

```
  input[type="submit"] {
  display: block;
  margin:  0  auto;
  width:  100px;
  padding:  15px;
  border: none;
  border-radius:  8px;
  background-color: darkslategray;
  color: white;
  cursor: pointer;
  font-size:  18px;
}

body {
  background-image: url(one.jpg);
  background-size: cover;
  background-repeat: no-repeat;
}

main {
  text-align: center;
  padding: 50px;
}

h2 {
  font-size:  25px;
  color: rgb(218,  216,  216);
}
```

8.2.3 prdiction_css.css -

```
body {
  background-image: url('image.jpg');
  background-size: cover;
  height: 100vh;
  display:  flex;
  align-items: flex-start;
  justify-content: flex-start;
```

```
  font-size: 46px;
  color: white;
  font-family: Arial, sans-serif;
  text-shadow: 2px 2px #dc7932;
}
```

8.3 Crop_app.py –

```
import joblib
from flask import Flask, render_template, request

app = Flask(__name__)

# Load the model when the Flask app starts
model = joblib.load('crop_app.pkl')

@app.route('/')
def home():
    return render_template('Home_1.html')

@app.route('/Predict')
def prediction():
    return render_template('Index.html')

@app.route('/form', methods=["POST"])
def brain():
    Nitrogen = float(request.form['Nitrogen'])
    Phosphorus = float(request.form['Phosphorus'])
    Potassium = float(request.form['Potassium'])
    Temperature = float(request.form['Temperature'])
    Humidity = float(request.form['Humidity'])
    Ph = float(request.form['ph'])
    Rainfall = float(request.form['Rainfall'])

    values = [Nitrogen, Phosphorus, Potassium, Temperature, Humidity, Ph,
Rainfall]

    if Ph > 0 and Ph <= 14 and Temperature < 100 and Humidity > 0:
        arr = [values]
        acc = model.predict(arr)
        return render_template('prediction.html', prediction=str(acc[0]))
    else:
        return "Sorry... Error in entered values in the form. Please check the
values and fill it again."

if __name___ == '__main__':
    app.run(debug=True)
```

8.5 IMPLEMENTING MACHINE LEARNING ALGORITHMS -

8.5.1 crop_prediction_app (1).ipynb :

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
```

```
[2]: df = pd.read_csv('Crop_recommendation.csv')
     df.head()
```

[2]:

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

```
[3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            2200 non-null   int64
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

```
[4]: if df['N'].all()>90:
         print(df['N'])
```

[5]:
```
df.isnull().sum()
```

[5]:
```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

[6]:
```
x = df.drop('label', axis = 1)
y = df['label']
```

[7]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, stratify = y, random_state = 1)
```

Testing Logistic Regression:

- The logistic regression model correctly predicted 96.18% of the test samples.
- An accuracy score of 0.9618 means that out of all the predictions made by the model, 96.18% were correct. This indicates high predictive performance for this specific dataset and task.

```
[8]: from sklearn.linear_model import LogisticRegression
     model = LogisticRegression()
     model.fit(x_train, y_train)
     y_pred = model.predict(x_test)
     from sklearn.metrics import accuracy_score
     logistic_acc = accuracy_score(y_test, y_pred)
     print("Accuracy of logistic regression is " + str(logistic_acc))

     Accuracy of logistic regression is 0.9618181818181818
```

Testing Decision Tree:

- **High Prediction Accuracy**: The decision tree model correctly predicted 97.82% of the test samples. This indicates that approximately 98 out of 100 predictions were accurate.
- **Effective Model Performance**: A high accuracy score demonstrates that the decision tree model effectively captures the relationship between the features (x_train) and the target labels (y_train).
- **Reliable and Generalizable**: The model's high accuracy on test data suggests good generalization ability, making it reliable for real-world applications.

```
[9]: from sklearn.tree import DecisionTreeClassifier
     model_2 = DecisionTreeClassifier(criterion='entropy',max_depth = 6, random_state = 2)
     model_2.fit(x_train, y_train)
     y_pred_2 = model_2.predict(x_test)
     decision_acc = accuracy_score(y_test, y_pred_2)
     print("Accuracy of decision  tree is " + str(decision_acc))

     Accuracy of decision  tree is 0.9781818181818182
```

Testing Naive Bayes :

- **Very High Prediction Accuracy**: The Naive Bayes model correctly predicted 99.45% of the test samples. This means that approximately 99 out of 100 predictions were accurate
- **Exceptional Model Performance**: A very high accuracy score indicates that the Naive Bayes model effectively captures the underlying patterns in the data, making accurate predictions.
- **Reliable and Generalizable**: The model's high accuracy on the test data suggests it generalizes well to unseen data, making it reliable for practical applications.

```
[10]: from sklearn.naive_bayes import GaussianNB
model_3 = GaussianNB()
model_3.fit(x_train, y_train)
y_pred_3 = model_3.predict(x_test)
naive_bayes_acc = accuracy_score(y_test, y_pred_3)
print("Accuracy of naive_bayes is " + str(naive_bayes_acc))

Accuracy of naive_bayes is 0.9945454545454545
```

Testing Random Forest :

- **Robust and Flexible**: Random Forest models are robust to overfitting, especially with a reasonable number of trees, and can handle large datasets with high dimensionality effectively.
- **High Suitability for Deployment**: Given its exceptional accuracy, the Random Forest model is highly suitable for deployment or further fine-tuning in real-world scenarios.
- **Feature Importance**: Random Forest models also provide insights into feature importance, which can be valuable for understanding which features are most influential in the predictions.

```
[11]: from sklearn.ensemble import RandomForestClassifier
model_4 = RandomForestClassifier(n_estimators = 25, random_state=2)
model_4.fit(x_train.values, y_train.values)
y_pred_4 = model_4.predict(x_test)
random_fore_acc = accuracy_score(y_test, y_pred_4)
print("Accuracy of Random Forest is " + str(random_fore_acc))

Accuracy of Random Forest is 0.9945454545454545
```

```
[18]: import pickle
```

```
[19]: Pkl_Filename = "Pickle_RL_Model.pkl"

      with open(Pkl_Filename, 'wb') as file:
          pickle.dump(model_4, file)
```

```
[20]: with open(Pkl_Filename, 'rb') as file:
          Pickled_Model = pickle.load(file)

      Pickled_Model
```

```
[20]: ▾ RandomForestClassifier
      RandomForestClassifier(n_estimators=25, random_state=2)
```

```
[21]: import sklearn
      print(sklearn.__version__)

      1.2.2
```

```
[31]: model = RandomForestClassifier()
      model.fit(X_train, y_train)
```

```
[32]: joblib.dump(model, 'crop_app.pkl')

[32]: ['crop_app.pkl']
```

```
[34]: from sklearn.ensemble import RandomForestClassifier
```

```
[35]: from sklearn.datasets import load_iris
```

```
[36]: from sklearn.model_selection import train_test_split
```

```
[38]: model = RandomForestClassifier()
      model.fit(x_train, y_train)
```

```
[38]: ▾ RandomForestClassifier
      RandomForestClassifier()
```

```
[39]: joblib.dump(model, 'crop_app.pkl')
```

```
[39]: ['crop_app.pkl']
```

8.5.2 Comparison of all the tested algorithms

Algorithm	Description	Accuracy	Differences and Key Points
Logistic Regression	A linear model for binary classification problems.	0.9618	Simple and interpretable, but may not capture complex relationships as well as non-linear models.
Decision Tree Classifier	A tree-based model using entropy for splits, max depth of 6.	0.9782	Easy to interpret and visualize, but prone to overfitting if not pruned properly.
Naive Bayes (GaussianNB)	Probabilistic classifier assuming Gaussian distribution of features.	0.9945	Fast and effective for certain problems, especially where feature independence assumption holds.
Random Forest Classifier	Ensemble of 25 decision trees with random feature selection.	0.9945	Combines multiple trees to improve accuracy and robustness, provides feature importance insights.

9. RESULTS

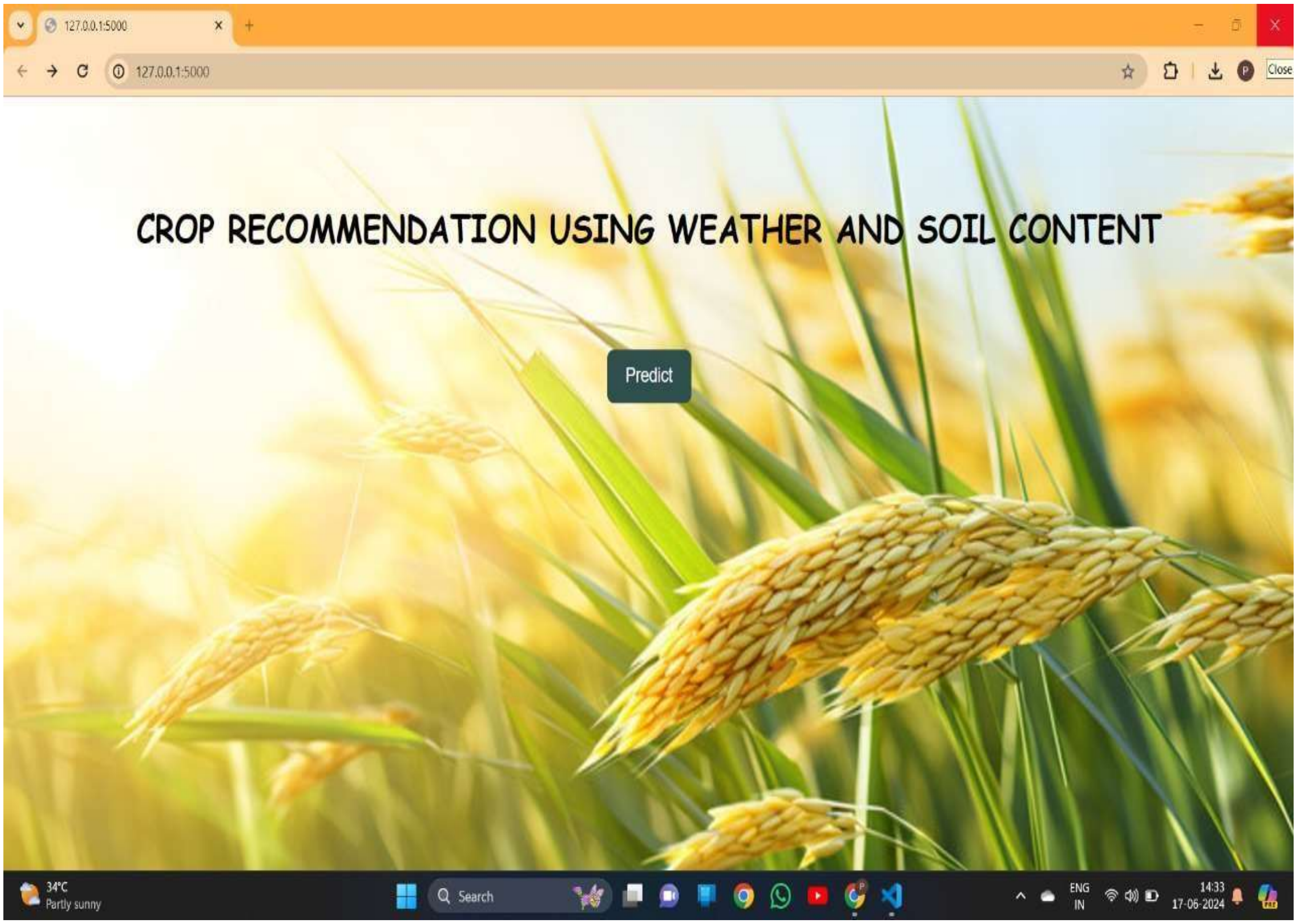

Fig 9.1 Home page

- This is the Home page of our website.
- Here we can navigate through the “Predict” button in order to enter the soil nutrients and weather conditions values.
- This home page is designed using the HTML and for styling purpose we have used the CSS.

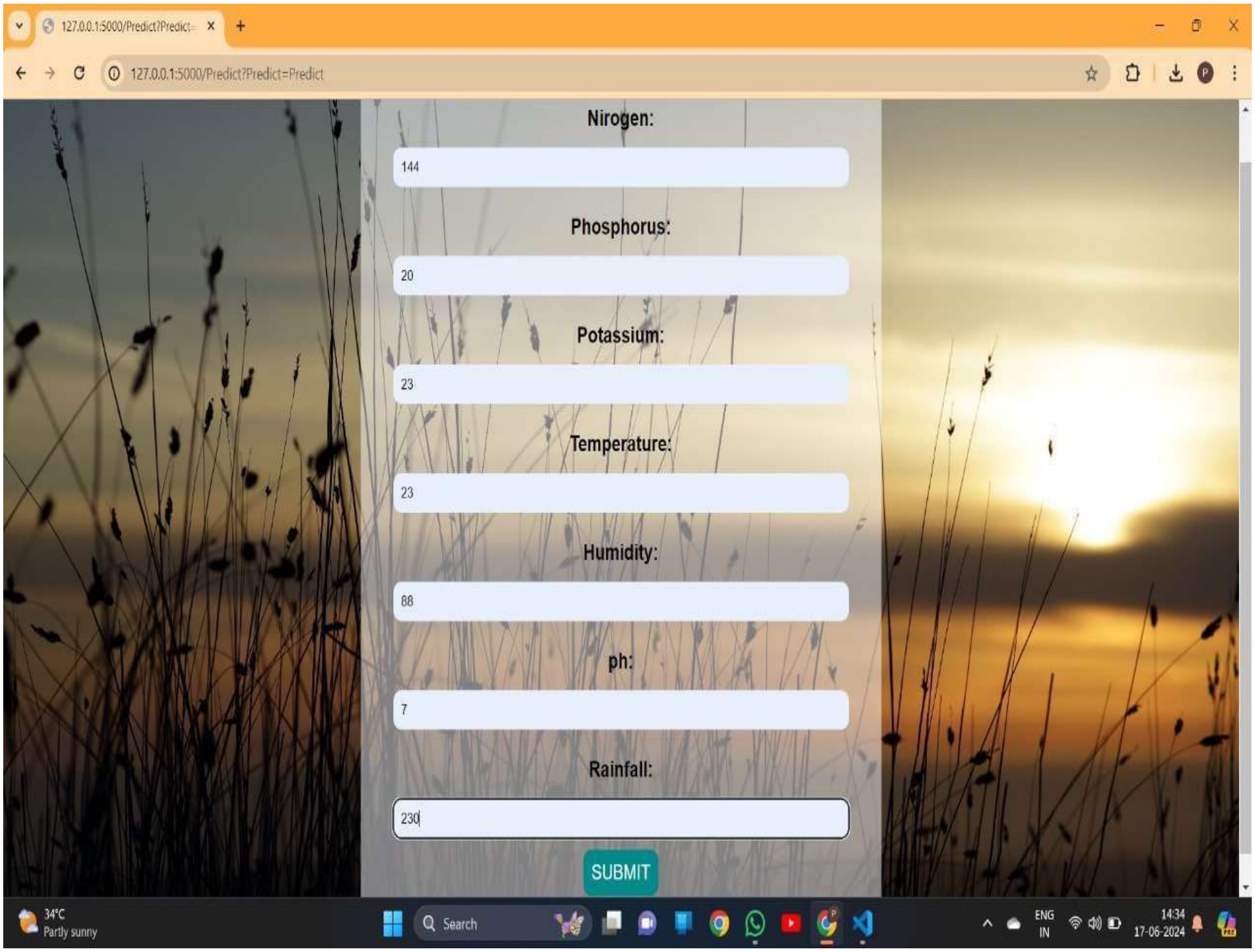

Fig 9.2 Index page

- This is the Index Page of our project.
- Here we have made a form using the <form> tag of HTML
- This page is also made up of HTML and CSS.
- In the given form, one must enter the values of the soil nutrients like Nitrogen, Phosphorus and Pottasium (NPK).
- We must also enter the values of the weather conditions like the temperature, humidity, ph value of soil and Rainfall in mm.
- After entering the values, if we press the submit button, we will be recommended with the required crop to be grown.

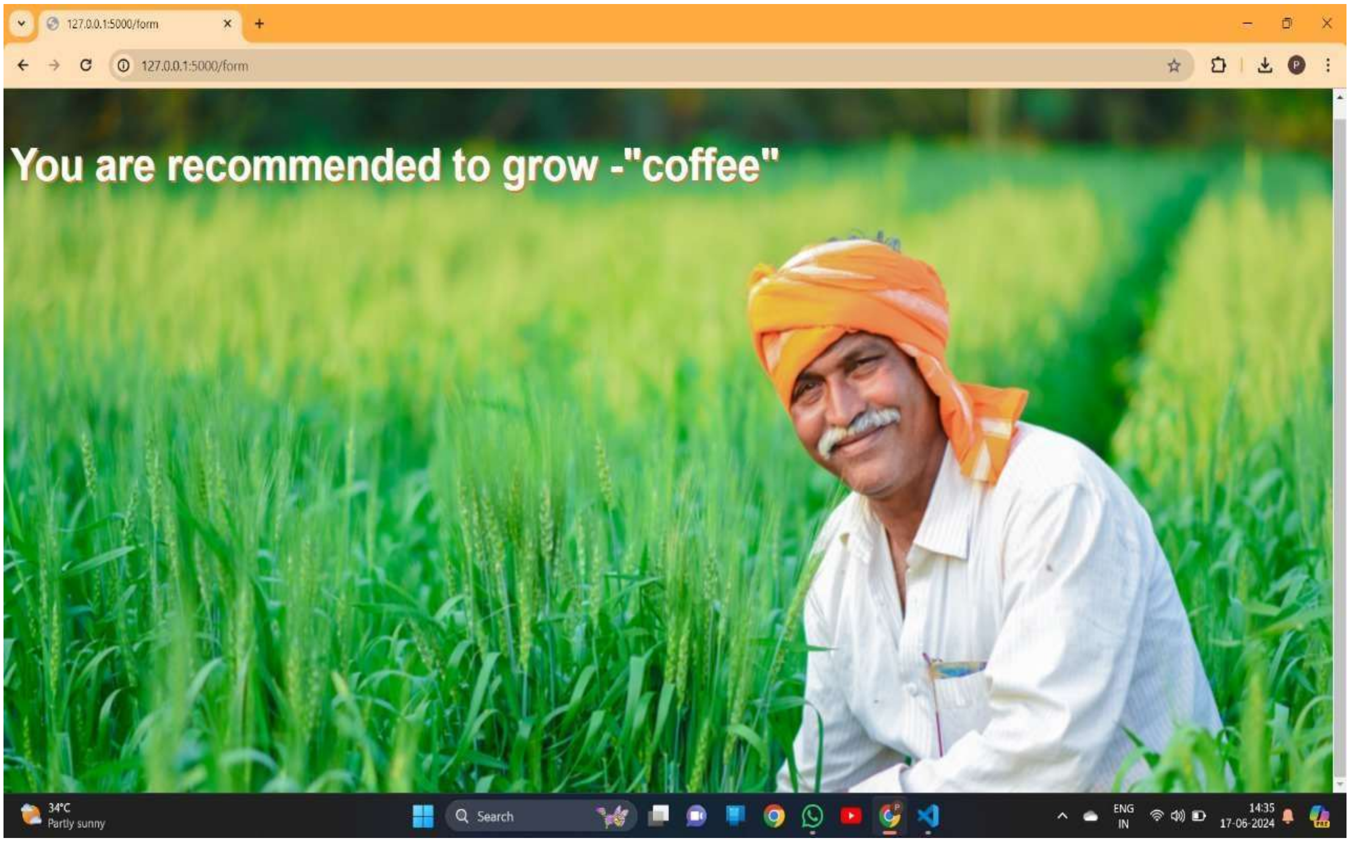

Fig 9.3 Prediction Image

- This is the final webpage of our project.
- Once we click on the submit button of the previous index page, we are going to be redirected to this page.
- This is being able to be possible as we are using a Machine Learning algorithm in the backend of our project.
- We tried few algorithms and at last chosen the algorithm which gave us the highest accuracy.
- This is going to help many farmers who wants to know which crop is best suitable for them to grow according to their soil nutrients and weather conditions.

10. CONCLUSION

Our Crop Recommendation System using Soil and Weather Conditions represents a groundbreaking advancement in the field of agricultural technology, embodying a vision for a brighter, more efficient future for farming communities.

By seamlessly integrating cutting-edge machine learning algorithms with real-time soil and weather data, our system offers farmers precise and actionable insights that transform crop selection from a guessing game into a science. This revolutionary approach not only enhances crop yield and farm productivity but also champions sustainable agricultural practices by ensuring that the right crops are cultivated under optimal conditions.

Harnessing the power of sophisticated algorithms such as Logistic Regression, Decision Tree, and Random Forest Classifier, our system consistently delivers high accuracy in crop recommendations. This has been rigorously validated through extensive testing and invaluable feedback from the very users it aims to serve.

In tackling the critical challenge of improving agricultural efficiency across diverse climatic conditions, our project holds particular significance in a country like India, where agriculture is not just an occupation but the backbone of the economy and a way of life for millions. By addressing the unique needs and challenges faced by Indian farmers, our system stands as a beacon of hope and progress.

We are profoundly confident that this system will bring about a substantial and positive impact on the agricultural sector. It empowers farmers to make informed decisions with confidence, paving the way for a more prosperous and sustainable agricultural landscape. This is more than just technology; it's a movement towards ensuring food security, economic stability, and a better quality of life for farmers and their communities.

Together, we can cultivate a future where innovation and tradition work hand in hand, fostering growth, resilience, and sustainability in agriculture.

REFERENCES

[1] Manpreet Kaur, Heena Gulati, Harish Kundra, “Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications”, International Journal of Computer Applications, Volume 99– No.12, August 2014.

[2] J. Meng, “Research on the cost of agricultural products circulation and its control under the new normal economic development,” Commercial Times, no. 23, pp. 145147, 2016.

[3] https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

[4] https://ieeexplore.ieee.org/document/9418351

[5] https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/1822-b.e-cse-batchno-336.pdf

[6] https://www.geeksforgeeks.org/crop-recommendation-system-using-tensorflow/