CropiGo - Crop Recommendation System

A PROJECT REPORT

by

ALAN D ANDOOR(VJC19CS013)
MATHEW K SHIBU(VJC19CS088)
NELVIN MATHEW(VJC19CS095)
ROSHAN SHAJI(VJC19CS107)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

MAY 2023

CropiGo - Crop Recommendation System

A PROJECT REPORT

by

ALAN D ANDOOR(VJC19CS013)
MATHEW K SHIBU(VJC19CS088)
NELVIN MATHEW(VJC19CS095)
ROSHAN SHAJI(VJC19CS107)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

under the guidance
of

Mrs. Arsha J K

Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

MAY 2023

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

Department of Computer Science and Engineering

Vision

Moulding socially responsible and professionally competent Computer Engineers to adapt to the dynamic technological landscape

Mission

1. Foster the principles and practices of computer science to empower life-long learning and build careers in software and hardware development.
2. Impart value education to elevate students to be successful, ethical and effective problem-solvers to serve the needs of the industry, government,society and the scientific community.
3. Promote industry interaction to pursue new technologies in Computer Science and provide excellent infrastructure to engage faculty and students in scholarly research activities.

Program Educational Objectives

Our Graduates

1. Shall have creative aid critical reasoning skills to solve technical problems ethically and responsibly to serve the society.
2. Shall have competency to collaborate as a team member and team leader to address social, technical and engineering challenges.
3. Shall have ability to contribute to the development of the next generation of information technology either through innovative research or through practice in a corporate setting.
4. Shall have potential to build start-up companies with the foundations, knowledge and experience they acquired from undergraduate education.

Program Outcomes

1. **Engineering knowledge**:Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**:Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / development of solutions**:Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**:Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**:Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:**Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:**Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:**Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:**Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to ones own work, as a member and unread in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**:Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

1. Ability to integrate theory and practice to construct software systems of varying complexity.

2. Able to Apply Computer Science skills, tools and mathematical techniques to analyse, design and model complex systems.

3. Ability to design and manage small-scale projects to develop a career in a related industry.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY

VAZHAKULAM, 686670

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"CropiGo - Crop Recommendation System"** is a bonafide work done by **ALAN D ANDOOR (VJC19CS013), MATHEW K SHIBU (VJC19CS088), NELVIN MATHEW (VJC19CS095)** and **ROSHAN SHAJI (VJC19CS107)** in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering from APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala during the academic year 2022-2023

Internal Supervisor **External Supervisor**

Project Coordinator **Head of Department**

ACKNOWLEDGEMENT

First and foremost, we thank **God Almighty** for His divine grace and blessings in making all these possible. May He continue to lead us in the years. It is our privilege to render our heartfelt thanks to our most beloved manager, **Rev. Msgr. Dr. Pius Malekandathil** , our director **Rev. Fr. Paul Nedumpurath** and our Principal **Dr. K K Rajan** for providing us the opportunity to do this project during the fourth year (2023) of our B.Tech degree course. We are deeply thankful to our Head of the Department, **Mr. Amel Austine** for his support and encouragement. We would like to express our sincere gratitude and heartfelt thanks to our project guide and coordinator **Mrs. Arsha J K** , Asst. Professor, Department of Computer Science and Engineering for her motivation, assistance and help for the project. We also thank all the staff members of the Computer Science Department for providing their assistance and support. Last, but not the least, We thank all our friends and family for their valuable feedback from time to time as well as their help and encouragement.

DECLARATION

We hereby declare that the project report entitled **“CropiGo - Crop Recommendation System”** submitted by us to the Department of Computer Science and Engineering, Viswajyothi College of Engineering and Technology, Vazhakulam, Muvattupuzha in partial fulfillment of the requirement for the award of the degree of B.Tech in Computer Science and Technology is a record of bonafide project work carried out by us under the guidance of Mr. Arsha J K. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this college.

Place: Vazhakulam

Date : ....

ALAN D ANDOOR

MATHEW K SHIBU

NELVIN MATHEW

ROSHAN SHAJI

ABSTRACT

Agriculture is a growing field of research. Crop prediction in agriculture is critical and depends upon the soil and environment conditions. In earlier days farmers were able to identify the crop suitable for the land and prediction of the yield was easier. Today rapid changes in environment made it difficult for the farmers to do so. The system takes into account various factors such as soil pH, temperature, rainfall, and other soil properties to determine the most appropriate crop to grow. The dataset contains 25 crops widely cultivated in Kerala. Users can select their village and run the prediction. If they need more precise information they can directly provide the tested soil properties as input for prediction. This work uses several machine learning algorithms such as Support Vector Machine (SVM), Random Forest (RF), Linear Regression (LR), Naive Bayes, and Decision tree. Among them, the Random Forest showed the best results with 91.4% accuracy. Additionally, the system also suggests the best season for cultivation and the fertilizers to boost up the yield.

Contents

List of Figures

List of Abbreviations

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
DT	Decision Tree
GUI	Graphic User Interface
HTML	Hypertext Markup Language
KNN	K Nearest Neighbor
LR	Linear Regression
ML	Machine Learning
MLR	Multivariate Linear Regression
RF	Random forest
SVM	Support Vector Machine

Chapter 1

INTRODUCTION

The population has dramatically expanded in recent years, which increases the need for food. The farming business needs to ramp up in order to fulfill the growing need for food, which begins with choosing the best suited crop to cultivate. Farmers used to be able to choose the crop best suited for their land and estimate the proper yield on their own. Yet due to the quick changes in the environment, it is no longer feasible for farmers to do so. That's where crop prediction using Machine Learning comes into play.

Crop prediction in agriculture is a complex process and several models have been suggested and tested to predict crops using these data. Crop cultivation depends on abiotic factors that can be classified into physical(temperature, humidity) and chemical(pH values,soil contents like nitrogen,phosphorus,potassium, etc.,) factors.

In this work we proposes a model that addresses the issues faced by farming community while choosing the best crops for cultivation.The Proposed system aims to provide a feasible solution to farmers to select crops on designated land based on soil, climatic and environmental conditions, and also suggest the best possible fertilizers to use.The system also provides data on proper growth management and optimal harvest timing.Users are provided with two choices of prediction. Users can directly give soil properties as input or they can run the crop prediction by selecting their village.Thus the system makes it more easy for users to choose the best crop suitable for their land and this will subsequently help to meet the increasing demand for the food supplies.

1.1 Problem Statement

Crop Prediction in agriculture is critical and is chiefly contingent upon soil and environment conditions, including rainfall, humidity, and temperature. Rapid changes in environmental conditions have made it difficult for the farming community to decide on the crop to be cultivated, monitor its growth, and determine when it could be harvested.

1.2 Objective

The Proposed system aims to provide a feasible solution to the farmers for choosing crops on the specified land based on climatic and environmental conditions and also suggest best possible fertilizers to use and the best season for cultivation.

1.3 Scope

The proposed system provides a feasible User Interface which enable every farmers to access and use the product. It recommends the best crop to cultivate which will increase the yield and profit. The system also provides data on proper growth management and best time to harvest.The system will be helpful to people who depends on agriculture as their source of income.

Chapter 2

LITERATURE SURVEY

2.1 Crop Recommender System Using Machine Learning Approach

Agriculture and its allied sectors are undoubtedly the largest providers of livelihoods in rural India. The agriculture sector is also a significant contributor factor to the country's Gross Domestic Product (GDP). However, regrettable is the yield per hectare of crops in comparison to international standards. This paper proposes a viable and user-friendly yield prediction system for the farmers. The proposed system provides connectivity to farmers via a user-friendly mobile application to recommend the most profitable crop. GPS helps to identify the user location and retrieve the rainfall estimation at the given area. The user provides the area and soil type as input. Machine learning algorithms allow choosing the most profitable crop list or predicting the crop yield for a user-selected crop as well as provide suggest the right time for using fertilizers.

2.1.1 Collecting the Raw Data

The data from previous years are the key elements in forecasting current performance. Historical data is collected from various reliable sources like data.gov.in, kaggle.com, and indianwaterportal.com. The data sets are collected for Maharashtra and Karnataka regions. The data has various attributes like state, district, year, season, type of crop, an area under cultivation, production, etc. The soil type is an attribute in other datasets with state and districts specification. This soil type column is extracted and merged into the main data set. Similarly, temperature and average rainfall are taken from a separate dataset and added to the main data sets for the specific region.

2.1.2 Data Preprocessing

The data sets are cleaned and pre-processed. The null values are replaced with mean values. The categorical attributes are converted into labels before processing the algorithms. The one hot encoding method is used to deal with categorical values in the data sets.

2.1.3 Training Algorithms

The machine learning approach is used for crop yield prediction. The patterns and correlations are discovered using ML approach. The Machine Learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), Multivariate Linear Regression (MLR), and K-Nearest Neighbour (KNN) are trained using historical data sets where the past experience is used to represent the outcome. Among the selected algorithms, the Random Forest regression provided the best accuracy.

2.1.4 Checking the accuracy over a training dataset

Among all the tested machine learning models, Random Forest algorithm proved to be the best for the given data set with an accuracy of 89%, followed by ML algorithms such as K Nearest Neighbor (KNN), Artificial Neural Network (ANN), Support Vector Machine (SVM) 75 and Multivariate Linear Regression (MLR) each with an accuracy of 87%, 86%, 75% and 60% repectively.

2.1.5 Setting up Mobile-App

As the next step the Mobile Application is designed and integrated with the best ML algoritm which is Random Forest here. Mobile Application offers multiple services. The farmer needs to register with the app through the registration process.Once the registration is complete, the farmer can use the mobile application services. The prediction module predicts the crop yield using the selected attributes from the data sets for the specific crop. The predict module also suggests the farmer with the highest yield crops. The fertilizer module guides the farmer for the right time to use the fertilizer.

2.1.6 Architecture Diagram

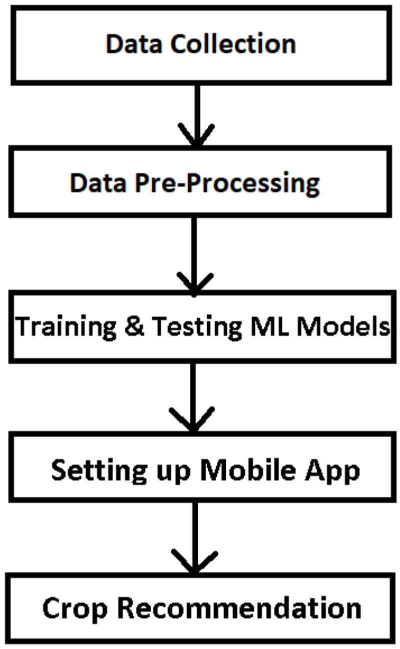

Figure 2.1: Architecture Diagram

2.1.7 Result

This paper highlighted a viable yield prediction system to the farmers while providing connectivity to farmers via a mobile application. The mobile application includes multiple features that users can leverage for the selection of a crop. The inbuilt predictor system helps the farmers to predict the yield of a given crop. The inbuilt recommender system allows a user exploration of the possible crops and their yield to take more educated decisions. The proposed model also explored the timing of applying fertilizers and recommends appropriate duration.

2.2 Crop prediction using machine learning

Crop prediction is one of agriculture's complex challenges, and several models have been developed and proven so far. Because crop production is affected by many factors such as atmospheric conditions, type of fertilizer, soil, and seed, this challenge necessitates using several datasets. This implies that predicting agricultural productivity is not a simple process; rather, it entails a series of complicated procedures. The paper aims to discover the best model for crop prediction, which can help farmers decide the type of crop to grow based on the climatic conditions and nutrients present in the soil. The project aims to compare various supervised learning algorithms like KNN, Decision Tree, and Random Forest on the dataset containing 22 varieties of crops.

2.2.1 Collecting the Raw Data

The 'Crop Recommendation' dataset is collected from the Kaggle website. The dataset takes into account 22 different crops as class labels and 7 features - Nitrogen content ratio , Phosphorus content ratio, Potassium content ratio in the soil, Temperature expressed in degree Celsius, Percentage of Relative Humidity, pH value and rainfall measured in millimeters.

2.2.2 Data Preprocessing

In this project the data processing method is to find missing values. Getting every data point for every record in dataset is tough. Empty cells, values like null or a specific character, such as a question mark, might all indicate that data is missing. The dataset used in the project didn't have any missing values

2.2.3 Train and Test Split

It is a process of splitting the dataset into a training dataset and testing dataset using traintestsplit() method of scikit learn module. 2200 data in the dataset has been divided as 80% of a dataset into training dataset-1760 and 20% of a dataset into testing dataset-440 data.

2.2.4 Fitting the model

Modifying the model's parameters to increase accuracy is referred to as fitting. To construct a machine learning model, an algorithm is performed on data for which the target variable is known. The model's accuracy is determined by comparing the model's outputs to the target variable's actual, observed values. Model fitting is the ability of a machine learning model to generalize data comparable to that with which it was trained. When given unknown inputs, a good model fit refers to a model that properly approximates the output.

2.2.5 Checking the score over a training dataset

Scoring, often known as prediction, is the act of creating values from new input data using a trained machine learning model. Using model.score() method calculating the score of each model over a training dataset shows how well the model has learned.

2.2.6 Predicting the model

When forecasting the likelihood of a specific result, “prediction” refers to the outcome of an algorithm after it has been trained on a previous dataset and applied to new data. Predicting the model using predict() method using test feature dataset. It has given the output as an array of predicted values.

2.2.7 Confusion Matrix and Classification Report

Confusion Matrix and Classification Report are the methods imported from the metrics module in the scikit learn library are calculated using the actual labels of test datasets and predicted values. Confusion Matrix gives the matrix of frequency of true negatives, false negatives, true positives and false positives. Classification Report is a metric used for evaluating the performance of a classification algorithm’s predictions. It gives three things: Precision, Recall and f1-score of the model. Precision refers to a classifier’s ability to identify the number of positive predictions which are relatively correct. It is calculated as the ratio of true positives to the sum of true and false positives for each class.

2.2.8 Architecture Diagram

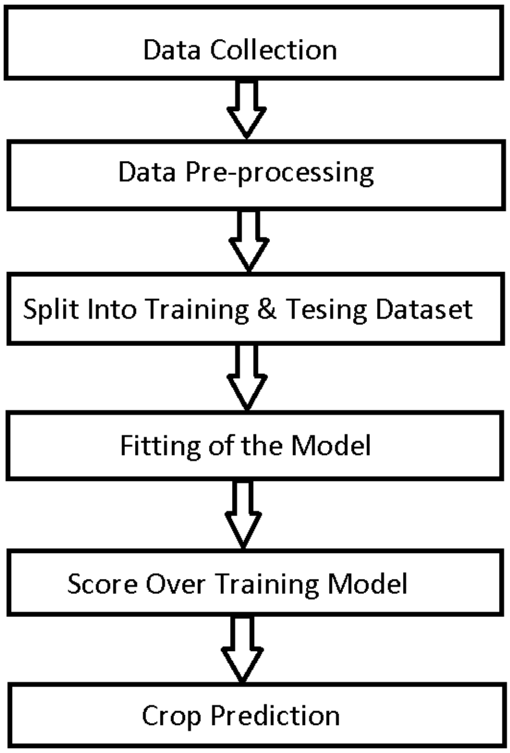

Figure 2.2: Architecture Diagram

2.2.9 Result

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 has the lowest accuracy 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.

2.3 Farm-Scale Crop Yield Prediction from Multi-Temporal Data Using Deep Hybrid Neural Networks

Farm-scale crop yield prediction is a natural development of sustainable agriculture, producing a rich amount of food without depleting and polluting environmental resources. Recent studies on crop yield production are limited to regional-scale predictions. For this research, creation of a large and reusable farm-scale crop yield production dataset is required, which could provide precise farm-scale ground-truth prediction targets. Therefore, multi-temporal data is utilized, such as Sentinel-2 satellite images, weather data, farm data, grain delivery data, and cadastre-specific data. A deep hybrid neural network model to train this multi-temporal data is introduced. This model combines the features of convolutional layers and recurrent neural networks to predict farm-scale crop yield production across Norway.

2.3.1 Multi-Temporal Data Collection and Data Handling

The primary data sources for the project are the official public archives of farmer grant applications and grain deliveries from the Norwegian Agriculture Agency. As Norwegian grain farmers rely on subsidies, they must fill out yearly grant applications describing the land used for crop cultivation.

Different datas collected for this research are:

1. Geographical Data
2. Weather Data
3. Satellite Image Data
4. Time Spans of Satellite and Weather Data
5. Grain delivery reports from farmers
6. Cultivated land: area and crop planted.

Here different datas are collected in the form of images and handled using image masking.Image masking makes it possible to focus on portions of an image that is of interest. Masks can be applied so that it either highlights certain parts or removes irrelevant parts from the image .

2.3.2 Data Preprocessing

i) Normalisation

Datas are acquired from various public resources, which must be normalised before training. Most features are normalised to fit a range between 0–1 better.

ii) Data Augmentation

The models proposed in this study are deep learning-based, which can be considerably data-hungry. Therefore, data augmentation techniques on the images were used to increase the overall dataset size and combat overfitting. Therefore three main data augmentation techniques were included: image cropping, image rotation, and random pixel noise.

iii) Identifying Crop Yield Prediction Targets

The proposed models presented in this section have a single output: the predicted crop yield per 1000 m^2. However, the target yield is slightly different between some models, requiring some clarification. Since the aim is to force the models to learn yield characteristics for each crop type.

2.3.3 Prediction Model Using Weather Data

Weather data directly include information for two of the four main factors of plant growth: precipitation and temperature. Training a deep neural network on the weather data allows us to verify the utilisation and relevancy of these data. Baseline Approach used here is the Weather DNN Model.

2.3.4 Prediction Models Using Satellite Images

Satellite images are remotely sensed data collected by earth-observing satellites. The multispectral satellite images contain detailed crop growth and plant health information, traditionally extracted using handcrafted vegetation indices. By training on per-farm satellite images, models can automatically extract important, relevant features for crop yield.

The models used in this work given below use raw satellite images to extract relevant features and make crop yield predictions.

1. Initial Experiment 1: The Single Image CNN Model
2. Initial Experiment 2: A Multi-Temporal CNN–RNN Model

2.3.5 Prediction Models Using Combined Satellite Images and Weather Data

Between the Weather DNN and multi-temporal CNN–RNN, all the previously described data are used as feature inputs. However, the models train and predict individually, so the models cannot learn any patterns that only appear when both satellite and weather data are combined. For this reason, the following models were created.

1. Initial Experiment: LSTM Model Using Handcrafted Features
2. Novel Approach 1: Pre-Trained Hybrid Model
3. Novel Approach 2: Hybrid CNN Model

2.3.6 Architecture Diagram

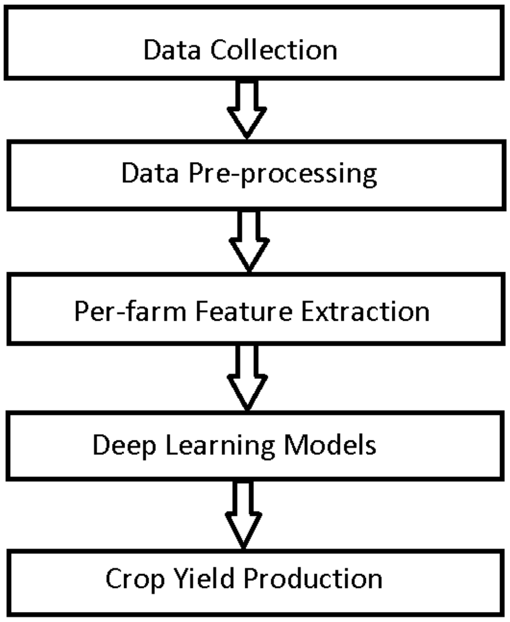

Figure 2.3: Architecture Diagram

2.3.7 Result and Discussions

Evaluating the initial experiments, the single image CNN can find a correlation between the multispectral satellite images and the yield of the farms.The best performing model is the hybrid CNN, which utilises both weather data and raw multispectral satellite images as its input and improves 8% over the baseline Weather DNN. The hybrid CNN can also make early, in-season predictions, though the error increases when data decreases.

In conclusion, this study explores and proves the following hypotheses:

1. Satellite images of farms and their surroundings can be used to accurately predict farm-scale crop yields.
2. Accurate field boundaries along with satellite images increase crop yield accuracy significantly.
3. Prediction accuracy can be further increased by combining satellite images and meteorological data.
4. It is possible to predict farm-scale crop yield earlier in the growing season with some reduced accuracy.

2.4 A Multiple Linear Regressions Model for Crop Prediction with Adam Optimizer and Neural Network Mlraonn

The proposed system is developed using real data with various soil parameters acquired from soil laboratory located in Chennai. This system uses 16 parameters of soil which includes all the micro, macro nutrients along with that pH, EC, OM values and the recommended crop for the soil parameter. The proposed Mlraonn (Multiple Linear Regression with Adam Optimization in Neural Network) model is developed using Keras software mainly used for Deep Learning. A neural network approach is used to construct a regression model.

2.4.1 Data Collection

Crop prediction with this proposed system developed with only by using soil properties such as micro; macro nutrients Ec; Om & pH values as input or independent features and suggested crop as output or dependent features. The above mentioned soil properties was collected from a soil lab. Dataset consists of nearly 1600 samples. The dataset is analyzed before generating a model.

2.4.2 Data Preprocessing

1) Finding Correlation Among Features:

In order to find a relationship between features a correlation map called heat map is generated. Heat map is used to find the correlation between each and every feature in the dataset. The correlation values ranges from -1 to +1; the correlation value of a feature which is near to -.01 to +.01 can be dropped since it denotes the value is equal to zero which mean there is no correlation.

2) Dataset Scaling

The dataset is examined to find out the range of the feature it is found that the values differ their exits no uniformity; with this; it is not possible to generate a correct model. A solution is to scale all the values in a predefined range which is nothing but -1 to +1.

3) Label Encoding

The target feature such as suggested crops seems to be of string data type; it is wise if it get converted in to numeric. The dataset has 27 different crop names; which comes under multiclass. Label encoding is a method which automatically assigns numerical value when it is called for a particular feature and the data type is converted in to integer array.

4) Handling Imbalanced Dataset

The value counts of the target variable in the dataset are found to be imbalanced.It is necessary to follow a technique which balances the dataset in order to get the correct prediction. SMOTE-Synthetic Minority Over Sampling Technique; in order to increase the sample size; synthetic data need to be generated; Smote uses KNN for oversampling the data.

2.4.3 Architecture Diagram

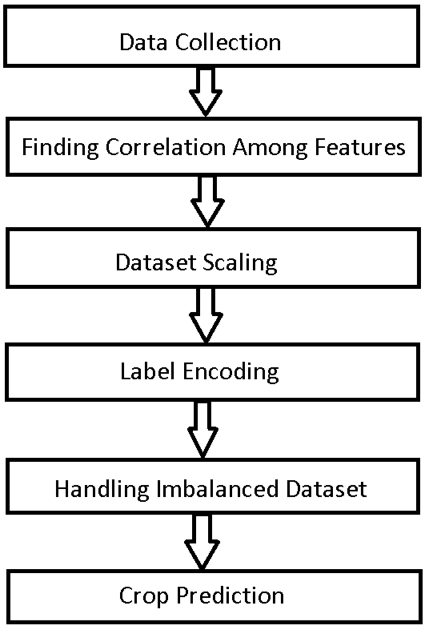

Figure 2.4: Architecture Diagram

2.4.4 Result

It is found that the result of the proposed Mlraonn model performed better with less epochs or iterations than the two standard machine learning algorithm.the proposed system also has some limitations which can also be considered as future enhancement; which are as follows; here the crop is predicted only by using the soil parameter. Other parameters like weather condition; wind speed, etc. is not included for the study. As a future enhancement more parameters other than the soil parameters can be included.

2.5 WB-CPI: Weather Based Crop Prediction in India Using Big Data Analytics

This work aims at collecting and analysing temperature, rainfall, soil, seed, crop production, humidity and wind speed data (in a few regions), which will help the farmers improve the produce of their crops. Firstly, data is preprocessed in a Python environment and then a MapReduce framework is applied, which further analyses and processes the large volume of data. Secondly, k-means clustering is employed on results gained from MapReduce and provides a mean result on the data in terms of accuracy. After that, bar graphs and scatter plots are used to study the relationship between the crop, rainfall, temperature, soil and seed type of two regions (Ahmednagar, Maharashtra and, Andaman and Nicobar Islands). Further, a self-designed recommender system has been used to predict the crops and display them on a GUI designed in a Flask environment.

2.5.1 Data Collection

Various data sets were collected during this step. Facing a little difficulty, seven datasets related to the workflow were found. and need from Kaggle and other websites.
Name of datasets:

1. crop_production.csv
2. INBOMBAY.csv
3. INCHENAI.csv
4. INCALCUT.csv
5. INDELHI.csv
6. portblair.csv
7. rainfall in India 1901-2015.csv

2.5.2 Data Pre-Processing

The collected datasets were combined and cleaned. The dataset was uploaded into the Colab notebook and pandas data frame were used to drop the useless columns and retain the ones important to us. NumPy, SciPy libraries carried out the calculations. A few index columns were added for future calculations. Interpolation, as represented below, was used to find the estimated value for the missing value in the dataset statistically.
dataframe_name.interpolate (method = 'linear' , direction = 'forward' , inplace = True)
where dataframe_name is the name of the dataset in use and the interpolation is done in the forward direction linearly. In the next step, the redundant and dirty data was eliminated using the

IQR and z-score method for detecting and deleting the outliers and interpolated the data in a few datasets where it was deemed fit.

2.5.3 Map Reduce

MapReduce model was implemented on the cleaned data, where the dataset is divided into key and column pairs. For the INBOMBAY.csv, INCHENAI.csv, INCALCUT.csv and INDELHI.csv, in the Map Function first, the month, year and region is taken as the key and the temperatures are taken as the value. Then in the Reduce Function calculation was performed to find the average monthly temperature of the regions and store the month, year and region as the key and the average temperature as the value. The year and region will be stored in the key and the respective parameter will be taken as the value month wise for the Map Function. In the Reduce function, these parameters will be calculated and assigned to crop seasons like Rabi, Kharif, Autumn, Winter, Summer, Whole Year, and the temporary December and November temperatures for the next year Rabi and Winter calculations. Then the Reduce function will calculate the produce per area for each row of data where the region, year, season and crop will form the key and produce per area will become the value. The produce per area of all the crops was found from the cropproduction.csv dataset, which will suggest a particular crop according to the region.

2.5.4 Python Recommendation Function

The cleaned and map reduced datasets of rainfall, temperature and crop production were combined along with the manually collected wind speed, humidity, soil type and seed type data to form one final dataset. Depending upon the season assigned, data is parsed to collect the best yield of the crops in the selected input region and state. Then depending upon the crop, the seed type and soil type with two different varieties are suggested as availability of the soil and the varieties of suitable and available seed types vary from region to region.

2.5.5 Website

A minimalistic website is designed as a graphical user interface for the user. The front end of the website has been made on Flask 2.0.0 in a virtual environment using Python 3, HTML 5.0, CSS3, Bootstrap 4 and Jinja2 templates. The website has four pages: Home, About, Contact, Solutions.

2.5.6 Visualization

Super dataset is used to find the relation between produce per area and a particular crop using a bar graph and scatter plot for a particular region.

2.5.7 Architecture Diagram

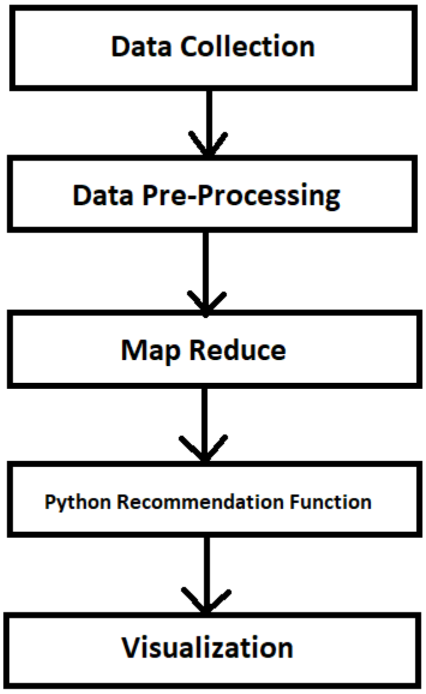

Figure 2.5: Architecture Diagram

2.5.8 Result

The proposed work introduces a crop recommendation system and uses MapReduce and K-means clustering, which gives efficient results in terms of computations.The model focuses on a wide range of crops and their produce per area along with the soil type and seed types depending on the varieties used in a particular region. From the visualisation graphs of K-Means clustering, the mean produce for a group of crops can be found.

Chapter 3

PROPOSED SYSTEM

The proposed system aims to help the farming community by making it easy for them to choose from a wide range of crops so that they get a better yield from it. The system predicts the best crops suitable for a land based on the agricultural environment of the land. In this work we used different classification algorithms to train a best model for crop prediction. Users can login or register in the webpage to avail the features. Once user is logged in they have access to crop prediction feature. Crop prediction feature provide two choices for the user: to predict the crop by selecting their village or by inputing the tested soil properties. In both cases a crop which is most suitable will be predicted along with the fertilizers to be used and the best season for cultivation. Users can ask their queries in the web page and check reply for them. Some useful links of government sites and details of agricultural offices are also provided.

3.1 Architecture Diagrams

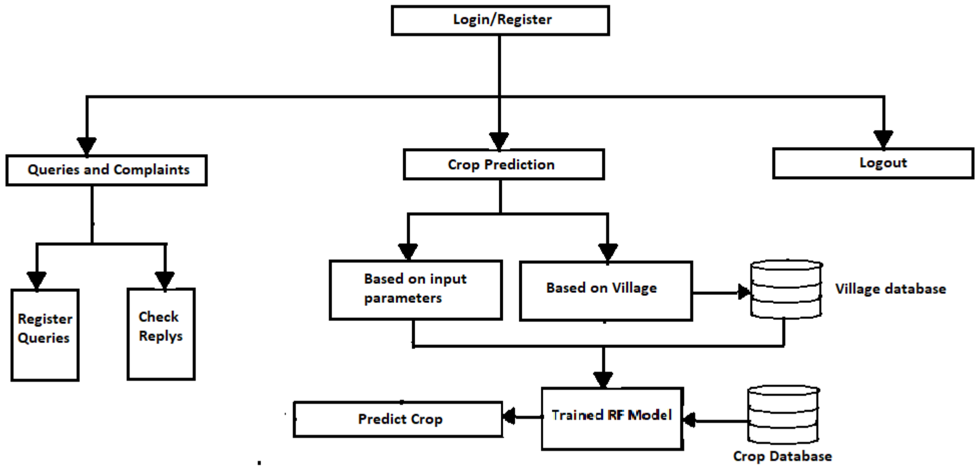

Figure 3.1: Architecture Diagram

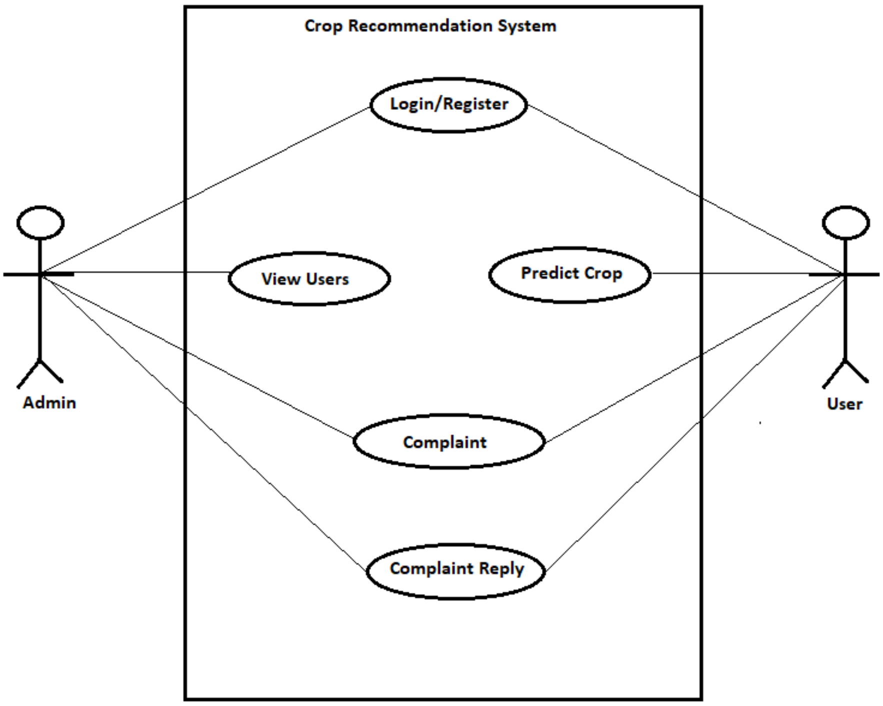

Figure 3.2: Usecase Diagram

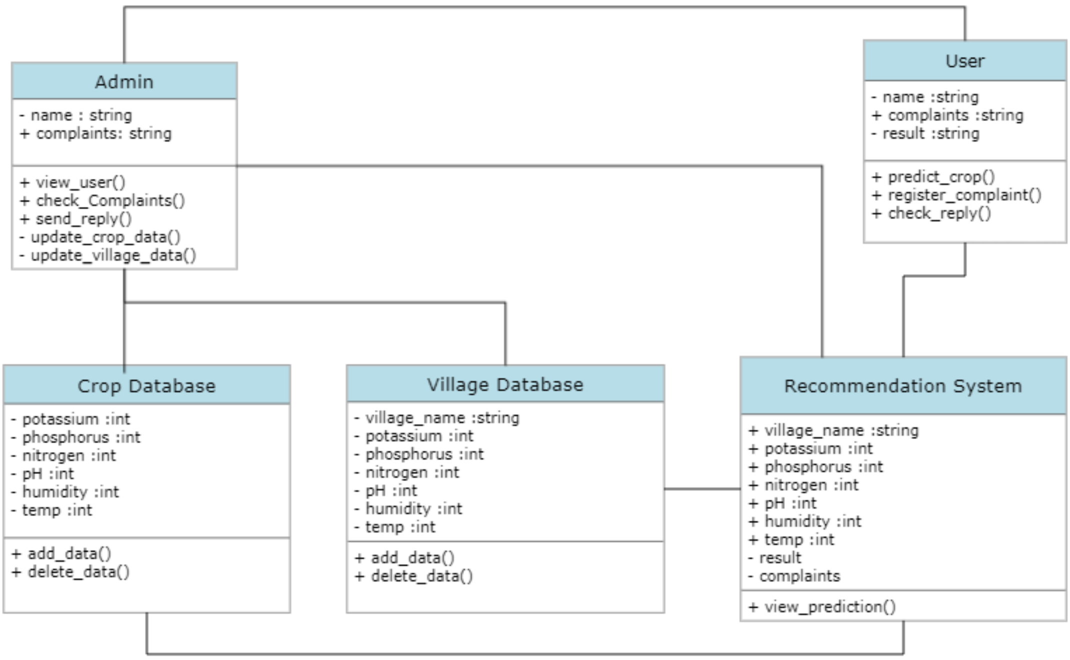

Figure 3.3: Class Diagram

Figure 3.4: Data Flow Diagram Level 0

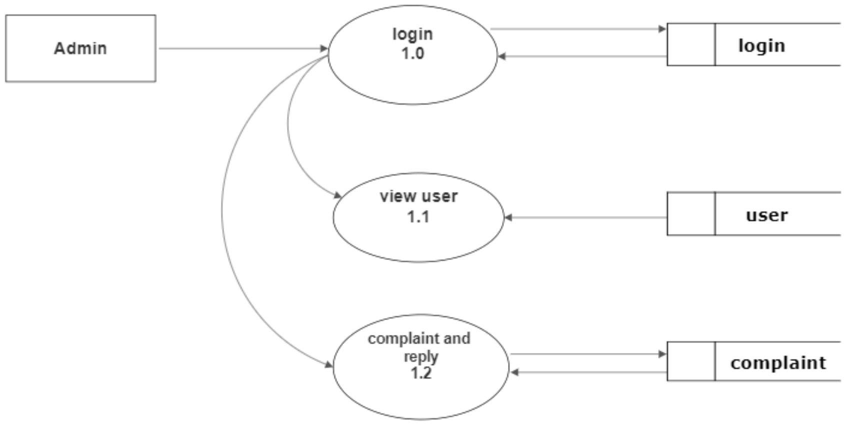

Figure 3.5: Data Flow Diagram Level 1 admin

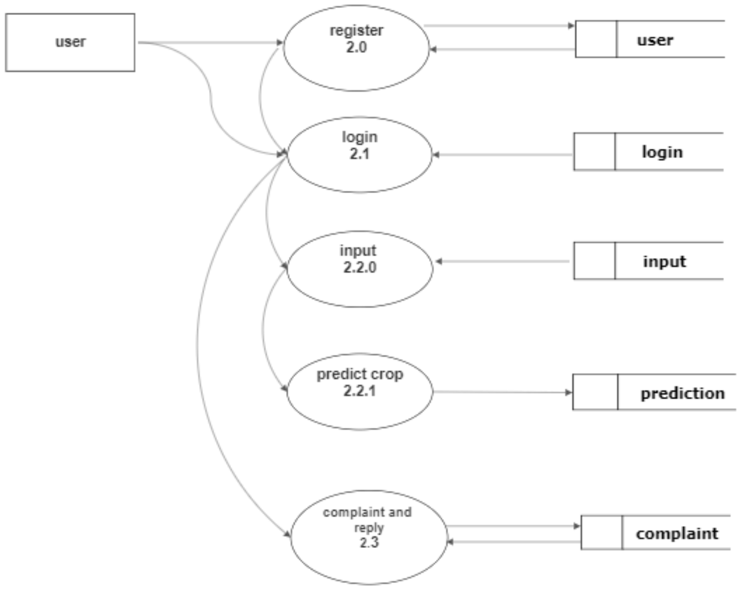

Figure 3.6: Data Flow Diagram Level 1 user

3.2 System Requirements

The basic requirements for the proposed systems can be divided into two.Hardware requirements and Software requirements.

3.2.1 Hardware Requirements

The hardware requirements consist of a laptop or personal computer having 4GB RAM or more having enough hard disk space.

3.2.2 Software Requirements

Coding is done in visual studio code and training of the classification model is done in google collab. SQLyog tool is used for creating database and tables for storing user data. The web application can be executed in any web browser.

3.2.3 Frameworks and Languages

The language mainly used for coding are HTML, Python, Javascript and jQuery. A python framework called flask is used for accessing tools and libraries required for building a web application.

Flask

Flask is a lightweight and flexible web framework for building web applications using Python. It is designed to be simple and easy to use, providing the essential features needed to develop web applications without imposing too many restrictions or unnecessary complexity.

jQuery

jQuery is a fast, lightweight, and feature-rich JavaScript library that simplifies web development by providing an easy-to-use API for interacting with HTML elements

Javascript

JavaScript is a versatile programming language primarily used for developing interactive and dynamic web applications. It is a high-level, interpreted language that can be executed on the client-side (in the web browser) as well as on the server-side (using platforms like Node.js).

3.3 Implementation Details

Our work mainly focuses on developing a crop recommendation model for kerala. The primary step of the project was to analyse previous works. All the previous works used different classification techniques for crop prediction. From analysis we identified that in those models users need to provide the soil properties like nitrogen, phosphorus, potassium, humidity, pH, temperature and rainfall as input. Our aim was to make the prediction more easier. So we created a ”village” dataset for our project which contains all villages in kerala and average soil properties corresponding to each village. So that once user select their village from the list the soil properties corresponding to the village can be used for prediction of the crop.

The next step was to create a web page for implementing the crop recommendation functionality.The web page is created using HTMl and Flask python framework. Further we needed to select a best classification model for predicting the crop. For that we trained our ”Crop recommendation” dataset using different machine learning algorithms such Decision Tree, Support Vector Machine (SVM), Linear regression (LR), Naive Bayes and random Forest. After comparing the results random Forest produced the best results.

The trained random forest model is converted into a pkl file and integrated into the web application. Input is collected from users through the web interface and crop prediction is done using the trained model. The interface is very simple so that people can use them easily.

3.4 Methodology

3.4.1 Data Collection

The proposed system needs two datasets named ”crop recommendation” and ”village”. The ”Crop recommendation dataset contains 25 different crops which widely cultivated in Kerala. Each crop has features such as nitrogen, phosphorus, potassium, humidity, temperature and pH. Each class of crop contains 100 rows of data. The data is mainly collected from kaggle data repository. Then the ”village” dataset contains all villages in kerala and average value of soil properties corresponding to each village. This dataset created manually based on the information obtained from the available government resources.

3.4.2 Data Preprocessing

In this step the unwanted or duplicate data from the dataset is eliminated from the ”Crop recommendation” dataset. The null values from the dataset is removed and the number of data rows for each crop is balanced for efficient performance of classication models. The village names in the ”village” dataset is converted to a format suitable for using while predicting the crop. The villages in the ”Village” dataset is classified based on their district for making the implementation process easier.

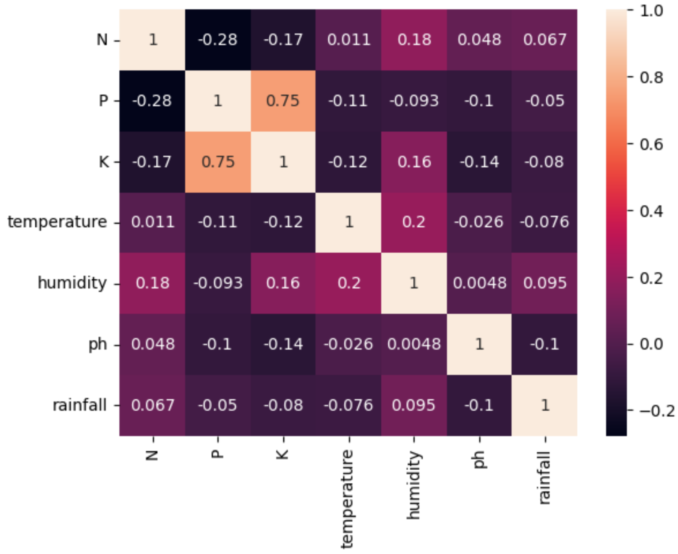

Figure 3.7: Heatmap of parameters

3.4.3 Model Training

The next step is to train the prepared dataset using different machine learning algorithms. We used Linear Regression(LR), Decision Tree, Support Vector Machine(SVM), Naive Bayes and Random Forest algorithms. Among these algorithms SVM produced the least accuracy. Logistic Regression produced an accuracy of 73.6 %. Decision Tree produced an accuracy 74.6 %. Random Forest produced the highest accuracy of 91.4 %. The trained Random Forest model is selected for crop prediction based on the test results.

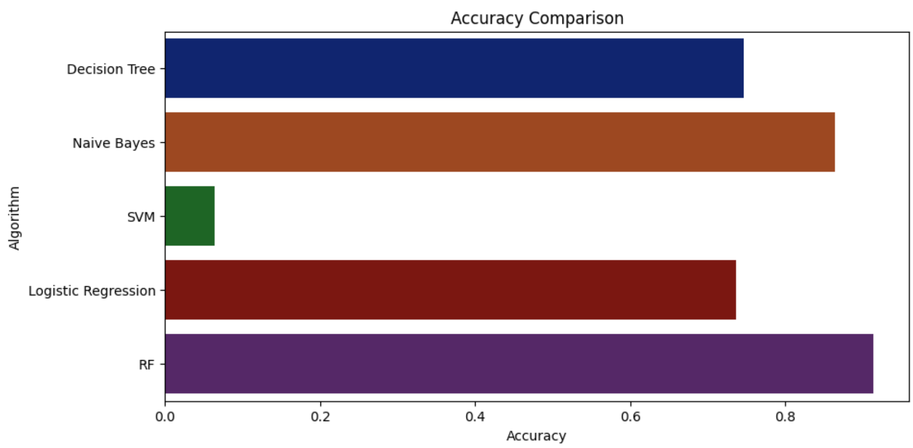

Figure 3.8: Accuracy Comparison

3.4.4 Crop Recommendation

The last step is to integrate the machine learning model into the web application. For that the trained Random forest model is converted into a python pickle file and used in the web application code. Crop prediction is done by using this trained model and the input parameters. For that the user has two choices provided in the web interface. User can directly input parameters or they can predict crop by selecting their village. The web page will output the crop predicted along with fertilizers suitable for the crop and the best season for cultivation.

Chapter 4

RESULT

The crop recommendation project yielded highly encouraging outcomes. The developed model accurately predicts the most suitable crops based on the available data, offering valuable insights for farmers. The recommendations provided by the model can assist farmers in optimizing their crop selection process, enabling them to make informed decisions that align with their specific agricultural conditions. With its notable accuracy and potential to enhance agricultural practices, this crop recommendation system stands as a valuable tool for farmers seeking improved productivity and profitability in their farming operations.

The designed model has an accuracy of 91.4 %.The system successfuly predicts the best crop for cultivating in a location based on the information provided by the user. The system also suggest the fertilizers to be used to increase the yield and also suggest the best season for cultivation.

Chapter 5

CONCLUSION

Predicting a suitable crop for cultivation is a challenging task, but the proposed Crop Recommendation System using Machine Learning is a powerful tool that can help farmers in making informed decisions about crop selection. The system uses data analysis and machine learning techniques to provide accurate and relevant recommendations for the farmers. The system can be easily deployed and accessed through a web-based interface, making it accessible to farmers from all over the world. To ensure the efficiency of the recommendations, the system uses the Random Forest model that gives the highest yield prediction accuracy among all the machine learning algorithms used. Also to ensure that the system will keep up its prediction accuracy in the future, it should be continuously updated with new data and feedback from farmers. The system, therefore, will not only assist farmers in selecting the right crop for the next season, but will also bridge the technical gap between agricultural and tech sectors. The crop recommendation project has successfully developed a robust and accurate model for suggesting suitable crops based on given data. The implemented solution provides valuable assistance to farmers in making informed decisions regarding crop selection, considering various factors such as soil characteristics and climate conditions. By leveraging this advanced technology, farmers can optimize their agricultural practices, enhance productivity, and make sustainable choices. The crop recommendation system holds great potential for transforming traditional farming approaches and empowering farmers to achieve greater efficiency and profitability in their operations.

References

[1] Shilpa Mangesh Pande, Dr. Prem Kumar Ramesh, Anmol, B.R Aiswarya, Karuna Rohilla and Kumar Shaurya, “Crop Recommender System Using Machine Learning Approach” in 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, doi:10.1109/ICCMC51019.2021.9418351.

[2] Madhuri Shripathi Rao, Arushi Singh, N.V. Subba Reddy and Dinesh U Acharya1, “Crop Prediction Using Machine Learning” in Journal of Physics: Conference Series, vol. 2161, no. 1, 2021, doi:10.1088/1742-6596/2161/1/012033.

[3] Martin Engen, Erik Sandø, Benjamin Lucas Oscar Sjølander, Simon Arenberg, Rashmi Gupta and Morten Goodwin, “Farm-Scale Crop Yield Prediction from Multi-Temporal Data Using Deep Hybrid Neural Networks”, Herbalism, vol. 1, no. 3, pp. 125–136, 2021.

[4] M. Lavanya and Dr. R. Parameswari, “A Multiple Linear Regressions Model for Crop Prediction with Adam Optimizer and Neural Network Mlraonn”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 4, 2020.

[5] Rishi Gupta, Akhilesh Kumar Sharma, Oorja Garg, Krishna Modi, Shahreen Kasim, Zirawani Baharum AND Salam A. Mostaoa, “WB-CPI: Weather Based Crop Prediction in India Using Big Data Analytics”, Int. J. Res. Appl. Sci. Eng. Technol., vol. 9, no. 5, pp. 189–193, October 2021, doi:10.1109/ACCESS.2021.3117247.

[6] Aruvansh Nigam, Saksham Garg, Archit Agrawal and Parul Agrawal ”Crop Yield Prediction Using Machine Learning Algorithms”, Fifth International Conference on Image Information Processing (ICIIP), 2019

[7] Manoj Kumar D P, Neelam Malyadri, Srikanth M S and Dr. Ananda Babu J, ”A Machine Learning model for Crop and Fertilizer recommendation”, Nat. Volatiles Essent. Oils, 2021.

[8] S. P. Raja, Barbara Sawicka, Zoran Stamenkovic and G. Mariammal, ”Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers”, IEEE Access, 2022, doi:10.1109/ACCESS.2022.3154350.

[9] Avinash Kumar, Sobhangi Sarkar and Chittaranjan Pradhan, ”Recommendation System for Crop Identification and Pest Control Technique in Agriculture”, International Conference on Communication and Signal Processing, April 4-6, 2019, India.

[10] Zeel Doshi, Subhash Nadkarni, Rashi Agrawal and Prof. Neepa Shah, ”AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms”, Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018.

Appendix A

Python code

A.1 Project Entry Part(main.py)

```
from flask import Flask
from public import public
from admin import admin
from user import user

app=Flask(__name__) #object for flask
app.secret_key = 'cropigo'

app.register_blueprint(public) #registering blueprint of public with main
app.register_blueprint(admin)
app.register_blueprint(user)

app.run(debug=True)#to run app
```

A.2 User Authentication Part(public.py)

```
from flask import Flask,Blueprint,render_template,request,session,url_for,redirect
from database import *
import re
public=Blueprint('public',__name__)

@public.route('/')
def home():
    return render_template('home.html')
```

```
@public.route(’/about’)
def about():
    return render_template(’about.html’)

@public.route(’/login’,methods=[’get’,’post’])
def login():
    if ’login’ in request.form:
        a=request.form[’username’]
        b=request.form[’password’]
        print(a,b)
        q="select * from login where user_name=’%s’ and password=’%s’"%(a,b)
        res=select(q)
        if res:
            print(res)
            login_id=res[0][’login_id’]
            if res[0][’user_type’]=="admin":
                session[’loggedin’] = True
                session[’name’] = res[0][’user_name’]
                return redirect(url_for(’admin.adminhome’))
            elif res[0][’user_type’]=="user":
                p="select * from user where user_name=’%s’ and login_id=’%s’"%(a,login_i
                res1=select(p)
                uname=res[0][’user_name’]
                upassword=res[0][’password’]
                session[’loggedin’] = True
                session[’name’] =res1[0][’full_name’]
                session[’loginid’]=res[0][’login_id’]
                session[’userid’]=res1[0][’user_id’]
                if uname==a and upassword==b:
                    return redirect(url_for(’user.userdashboard’))
        else:
            message="Error | Invalid username or password!"
            return render_template("login.html",message=message)
    return render_template(’login.html’)
```

```
@public.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('userid', None)
    session.pop('email', None)
    session.pop('name', None)
    return redirect('/')

@public.route('/signup',methods=['get','post'])
def signup():
    if 'signup' in request.form:
        name=request.form['name']
        email=request.form['email']
        username=request.form['username']
        password=request.form['password']
        place=request.form['place']
        phone_number=request.form['phone_number']
        querry="select * from login where user_name='%s'"%(username)
        result=select(querry)
        if result:
            message="Account Already Exists!"
            return render_template('signup.html',message=message)
        else:
            q="insert into login(user_type,user_name,password) values
                    ('user','%s','%s')"%(username,password)
            res=insert(q)
            q="insert into user(login_id,full_name,place,phone_number,email_id,
                    user_name) values('%s','%s','%s','%s','%s','%s')"%(res,
                    name,place,phone_number,email,username)
            insert(q)
            message="Account Created Succesfully!!"
            return render_template('signup.html',message=message)
    return render_template('signup.html')
```

A.3 User Functionality Part(user.py)

```
from flask import Flask,Blueprint,render_template,request, session,url_for,redirect
from datetime import date
from database import *
from sklearn.ensemble import RandomForestClassifier
import pickle
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
user=Blueprint('user',__name__)

@user.route('/userdashboard',methods=['get','post'])
def userdashboard():
    return render_template('userdashboard.html')

@user.route('/usercomplaints',methods=['get','post'])
def usercomplaints():
    if 'complaint-submit' in request.form:
        complaint=request.form['complaint-data']
        today=date.today() #for date
        querry="insert into complaint(user_id,complaint,reply,date) values
                ('%s','%s','pending','%s')"%(session['userid'],complaint,today)
        insert(querry)
    if 'reply-check' in request.form:
        q="select complaint,date,reply from complaint WHERE user_id='%s'"
                %(session['userid'])
        res=select(q)
        print(res)
        return render_template('usercomplaints.html',data=res)
    return render_template('usercomplaints.html')

@user.route('/croppredict',methods=['get','post'])
def croppredict():
    return render_template('croppredict.html')
```

```
@user.route('/village',methods=['get','post'])
def village():
    if 'crop-predict' in request.form:
        villagename=request.form.get('inputVillage')
        print(villagename)
        village_data= pd.read_csv("static\cropigovillage.csv")
        nitrogen=village_data[village_data['VILLAGES'] == villagename]['N'].values[0]
        phosphorus=village_data[village_data['VILLAGES'] == villagename]['P'].values[0]
        potassium=village_data[village_data['VILLAGES'] == villagename]['K'].values[0]
        temperature=village_data[village_data['VILLAGES'] == villagename]
                ['temperature'].values[0]
        humidity=village_data[village_data['VILLAGES'] ==villagename]
                ['humidity'].values[0]
        ph=village_data[village_data['VILLAGES'] == villagename]['ph'].values[0]
        rainfall=village_data[village_data['VILLAGES'] == villagename]
                ['rainfall'].values[0]
        RF_pkl_filename = 'static\RandomForest.pkl'
        with  open(RF_pkl_filename, 'rb') as RF_Model_pkl:
            RF=pickle.load(RF_Model_pkl)
        data = np.array([[nitrogen,phosphorus, potassium, temperature,
                humidity, ph, rainfall]])
        prediction = RF.predict(data)
        print(prediction)
        crop=prediction[0]
        print(crop)
        #adding imagpath
        con=''.join(prediction)
        image=con+'.jpg'
        fertilizer_season= pd.read_csv('static\cropfertilizer&season.csv')
        fertilizer = fertilizer_season[fertilizer_season['Crop']==crop]
                ['Fertilizer'].values[0]
        season = fertilizer_season[fertilizer_season['Crop']==crop]
                ['Season'].values[0]
        return render_template('/resultpage.html',crop=crop,
                fertilizer=fertilizer,season=season,path=image)
```

```
    return render_template('village.html')

@user.route('/gencropprediction',methods=['get','post'])
def gencropprediction():
        if 'crop-predict' in request.form:
            nitrogen=request.form.get('nitrogen',type=int)
            phosphorus=request.form.get('phosphorus',type=int)
            potassium=request.form.get('potassium',type=int)
           temperature=request.form.get('temperature',type=float)
            humidity=request.form.get('humidity',type=float)
            ph=request.form.get('ph',type=float)
            rainfall=request.form.get('rainfall',type=float)
            RF_pkl_filename = 'static\RandomForest.pkl'
            with  open(RF_pkl_filename, 'rb') as RF_Model_pkl:
                RF=pickle.load(RF_Model_pkl)
            data = np.array([[nitrogen,phosphorus, potassium, temperature,
                    humidity, ph, rainfall]])
            prediction = RF.predict(data)
            print(prediction)
            crop=prediction[0]
            print(crop)
            con=''.join(prediction)
            image=con+'.jpg'
            fertilizer_season= pd.read_csv('static\cropfertilizer&season.csv')
            fertilizer = fertilizer_season[fertilizer_season['Crop'] == crop]
                    ['Fertilizer'].values[0]
            season = fertilizer_season[fertilizer_season['Crop'] == crop]
                    ['Season'].values[0]
            return render_template('/resultpage.html',crop=crop,fertilizer=
                    fertilizer,season=season,path=image)
        return render_template('/gencropprediction.html')

@user.route('/resultpage',methods=['get','post'])
def resultpage():
    return render_template('/resultpage.html')
```

A.4 Admin Functionality Part(admin.py)

```
from flask import *
from database import *
admin=Blueprint('admin',__name__) #creating blueprint for admin page

@admin.route('/admin',methods=['get','post'])
def adminhome():
    if 'user-search' in request.form:
        name=request.form['user-search-name']
        q="select * from user where user_name='%s'"%(name)
        if name=='all':
            q="select * from user"
        res=select(q)
        print(res)
        return render_template('adminhome.html',data=res)
    if 'complaint-check' in request.form:
        c="select * from complaint where reply='pending'"
        cmp=select(c)
        return render_template('adminhome.html',complaint_data=cmp)
    if 'reply' in request.form:
        reply=request.form['reply-message']
        cid=request.form['complaintid']
        q="update complaint SET reply='%s' WHERE complaint_id='%s'"%(reply,cid)
        update(q)
    return render_template('adminhome.html')
```

A.5 Database Connectivity Part(database.py)

```
import mysql.connector
password=""
database ="user_data"
port=3306
def select(q):
cnx = mysql.connector.connect(user="root", password=password, host="localhost",
        database=database,port=port)
```

```
cur = cnx.cursor(dictionary=True)
cur.execute(q)
result = cur.fetchall()
cur.close()
cnx.close()
return result
def update(q):
cnx = mysql.connector.connect(user="root", password=password, host="localhost",
        database=database,port=port)
cur = cnx.cursor(dictionary=True)
cur.execute(q)
cnx.commit()
result = cur.rowcount
cur.close()
cnx.close()
return result
def delete(q):
cnx = mysql.connector.connect(user="root", password=password, host="localhost",
        database=database,port=port)
cur = cnx.cursor(dictionary=True)
cur.execute(q)
cnx.commit()
result = cur.rowcount
cur.close()
cnx.close()
def insert(q):
cnx = mysql.connector.connect(user="root", password=password, host="localhost",
        database=database,port=port)
cur = cnx.cursor(dictionary=True)
cur.execute(q)
cnx.commit()
result = cur.lastrowid
cur.close()
cnx.close()
return result
```

A.6 Machine Learning Part

```
from __future__ import print_function
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')
PATH = '/content/Crop_recommendation.csv'
df = pd.read_csv(PATH)
features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = df['label']
labels = df['label']
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,
        test_size = 0.2,random_state =2)
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)
predicted_values = RF.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x*100)
print(classification_report(Ytest,predicted_values))
import pickle
RF_pkl_filename = 'RandomForest.pkl'
RF_Model_pkl = open(RF_pkl_filename, 'wb')
pickle.dump(RF, RF_Model_pkl)
RF_Model_pkl.close()

#Making Prediction using Random Forest Classifier
data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
prediction = RF.predict(data)
print(prediction)
```

Appendix B

SCREENSHOTS

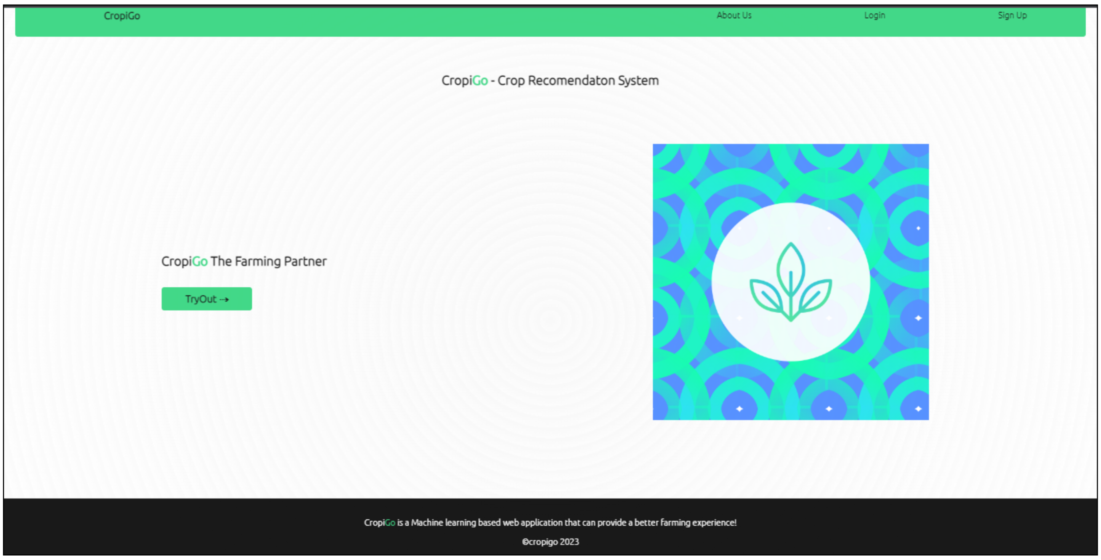

Figure B.1: Home page

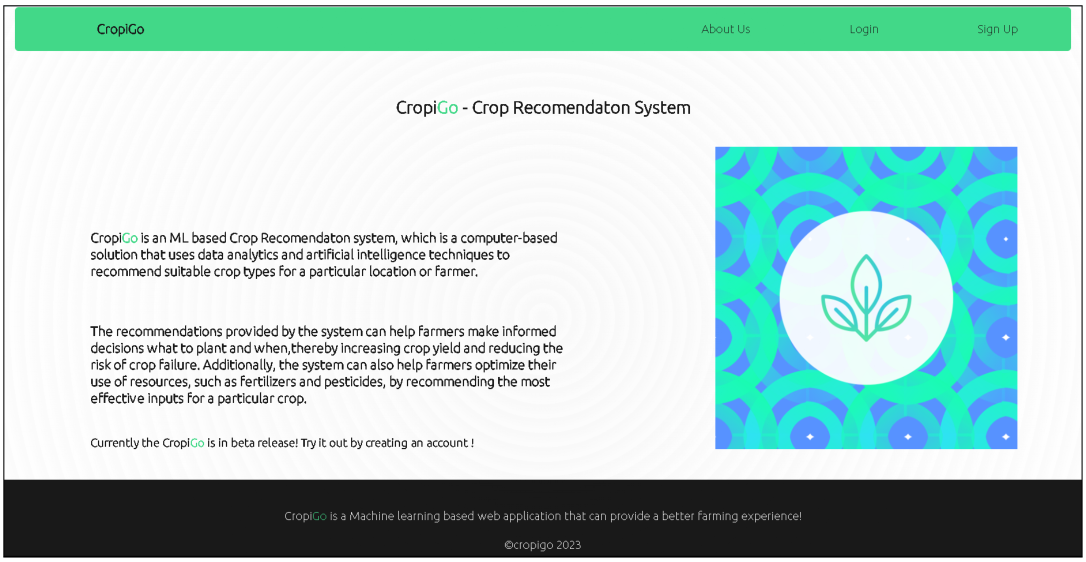

Figure B.2: About Us page

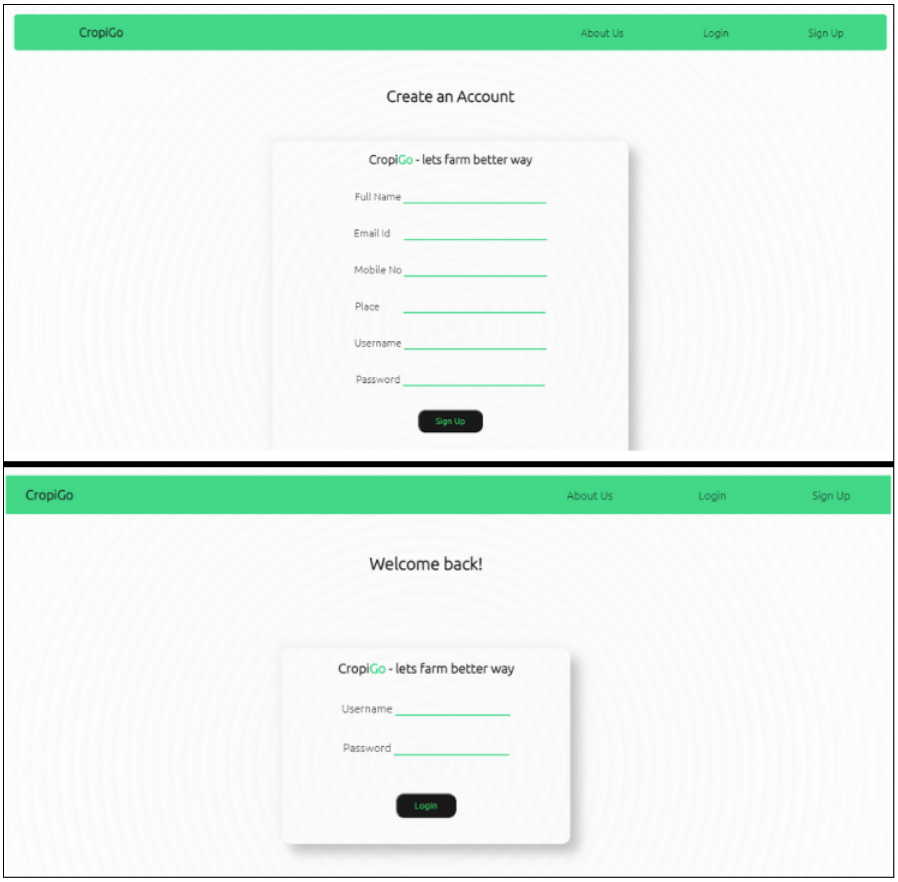

Figure B.3: Sign up and Login pages

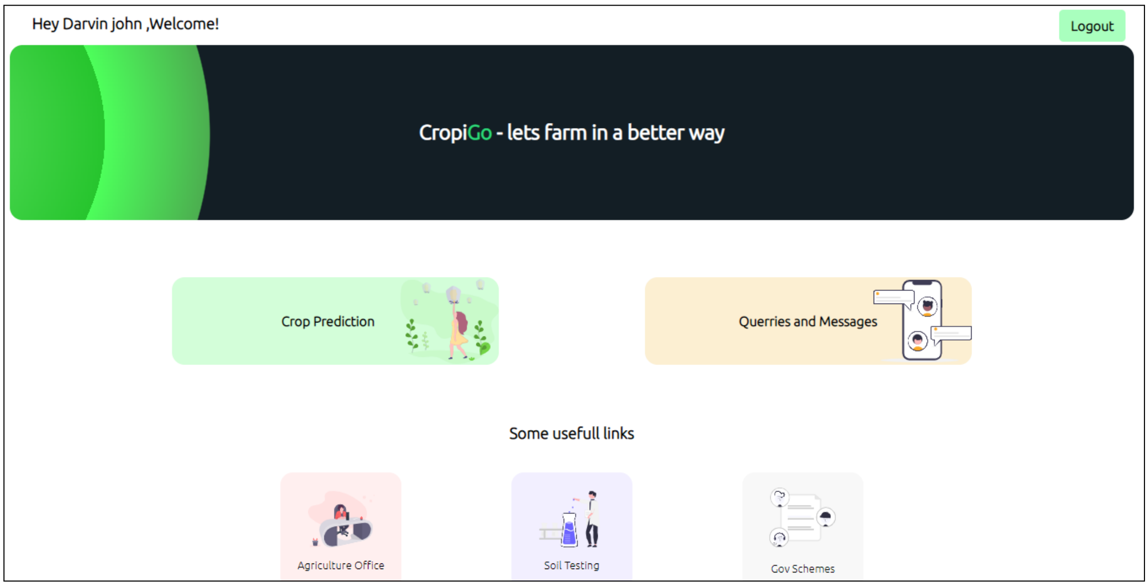

Figure B.4: User Dashboard

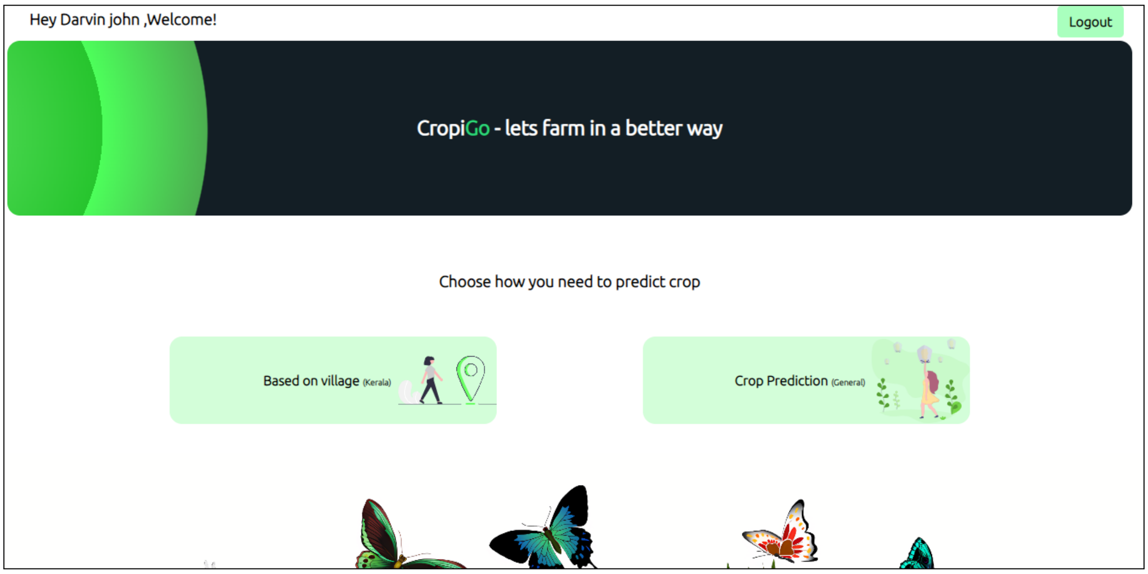

Figure B.5: Crop Prediction Page

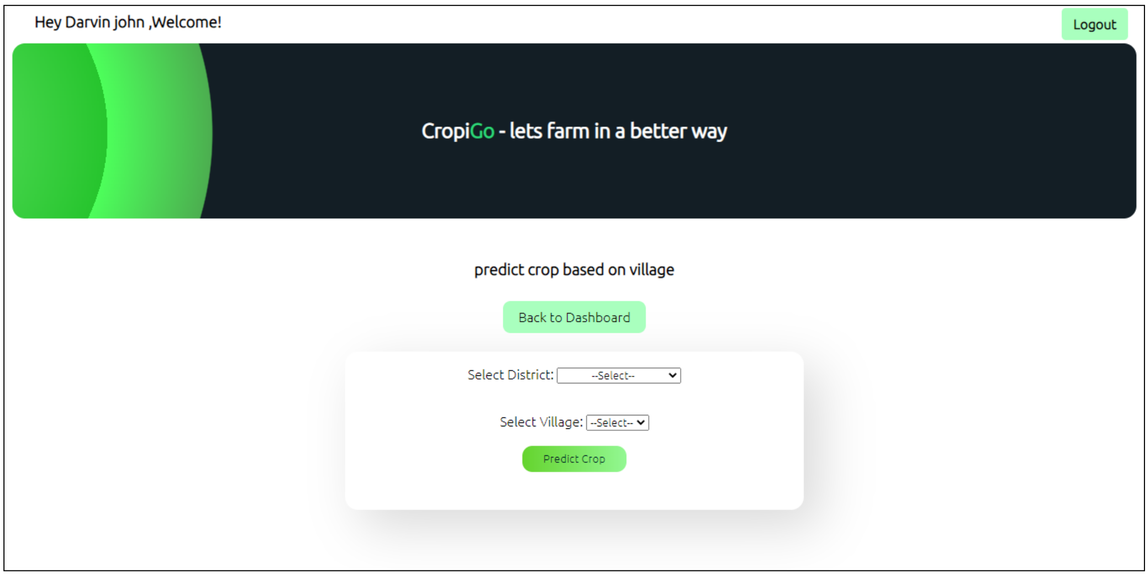

Figure B.6: Crop Prediction based on Village

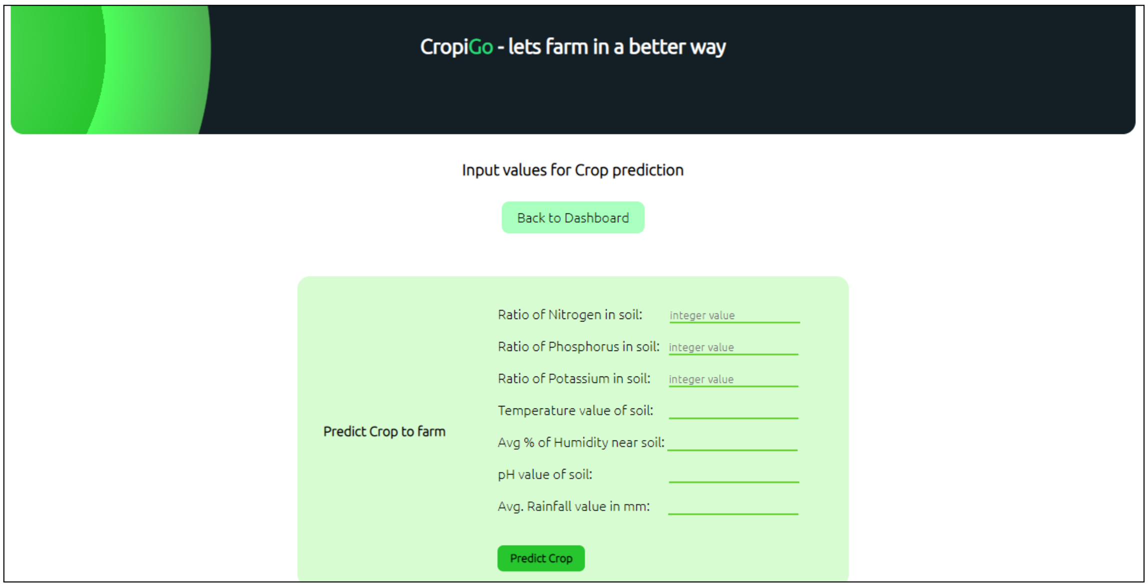

Figure B.7: Crop Prediction based on Soil Properties

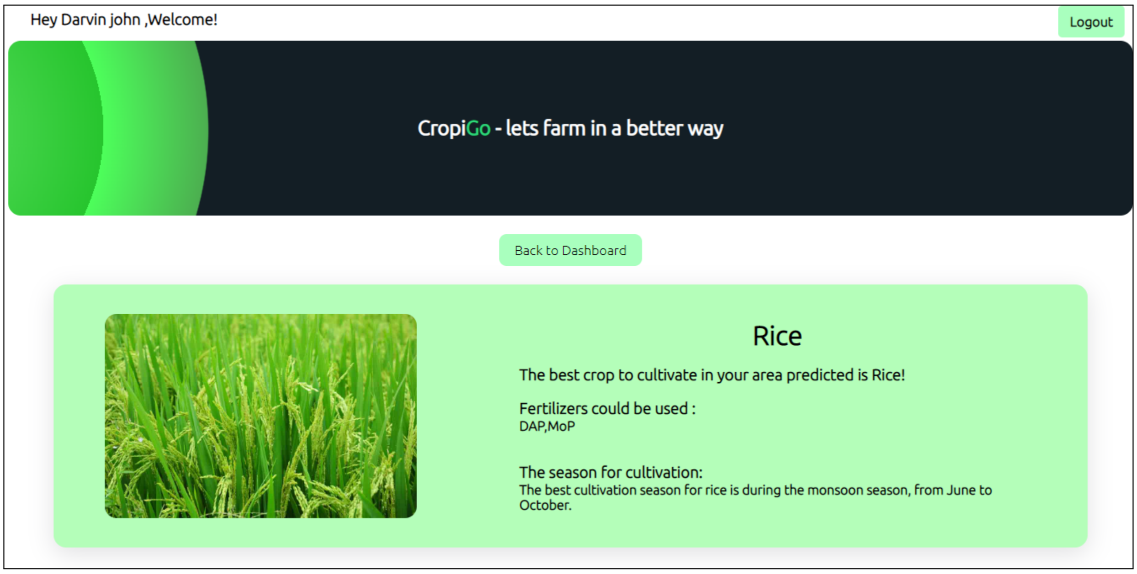

Figure B.8: Crop Recommendation Result