CROP YIELD PREDICTION USING MACHINE LEARNING

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

K. Shanmukh Jagan Chowdary	M. Sruthi Keerthi Harshni	V. Praveen Srikar	K. Shanmukha Krishna
20ME1A0593	20ME1A05A6	20ME1A0569	20ME1A0596

Under the Guidance of

Mr. Y Nagendra Kumar
Assistant Professor

Department of Computer Science and Engineering

RAMACHANDRA COLLEGE OF ENGINEERING (AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTUK, Kakinada)

Accredited by NBA, NAAC A+

NH-16 Bypass, Vatluru (V), Eluru -534007, Eluru Dist., A.P

2020 – 2024

RAMACHANDRA COLLEGE OF ENGINEERING (AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTUK, Kakinada)

Accredited by NBA, NAAC A+

NH-16 Bypass, Vatluru (V), Eluru -534007, Eluru Dist., A.P

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **K. Shanmukh Jagan Chowdary (20ME1A0593), M. Sruthi Keerthi Harshni (20ME1A05A6), V. Praveen Srikar (20ME1A0569), K. Shanmukha Krishna (20ME1A0596)** students of **Bachelor of Technology in Computer Science & Engineering** have successfully completed their project work entitled **"CROP YIELD PREDICTION USING MACHINE LEARNING"** at **Ramachandra College of Engineering, Eluru during the Academic Year 2023-2024.** This document is submitted in partial fulfillment for theaward of the Degree of Bachelor of Technology in Computer Science & Engineering and the same is not submitted elsewhere.

Mr. Y Nagendra Kumar **Project Guide**	**Dr. G Chamundeswari** **Professor & HOD-CSE**

External Examiner

DECLARATION

We are **K. Shanmukh Jagan Chowdary (20ME1A0593), M. Sruthi Keerthi Harshni (20ME1A05A6), V. Praveen Srikar (20ME1A0569), K. Shanmukha Krishna (20ME1A0596)** hereby declares the project report titled "**CROP YIELD PREDICTION USING MACHINE LEARNING**" under the supervision of **Mr. Y Nagendra Kumar, Asst Professor** Department of Computer Science and Engineering is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering.

This is a record of work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

ACKNOWLEDGEMENT

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our project work. It is our pleasure andresponsibility to acknowledge the help of all those individuals.

We sincerely thank our guide **Mr. Y Nagendra Kumar, Asst Professor** in the Department of CSE for helping us in successful completion of our project under her supervision.

We are very grateful to **Dr. G Chamundeswari**, Head of the Department, Department of Computer Science & Engineering for her assistance and encouragement in all respects in carrying throughout our project work.

We express our deepest gratitude to **Dr. S. Subrahmanya Sarma,** i/c Principal, Ramachandra College of Engineering, and Eluru for his valuable suggestions during preparation of draft in our document.

We thank the Project Coordinator **Dr. A Daveedu Raju, Professor,** Department of CSE, for his valuable guidance and support throughout the development of this project.

We express our deepest gratitude to **the Management** of Ramachandra College of Engineering, Eluru for their support and encouragement in completing our project work and providing us necessary facilities.

We sincerely thank all the faculty members and staff of the Department of CSE for their valuable advices, suggestions and constant encouragement which played a vital role in carrying out this project work.

Finally, we thank one and all who directly or indirectly helped us to complete our project work successfully.

K. Shanmukh Jagan Chowdary	**M. Sruthi Keerthi Harshni**	**V. Praveen Srikar**	**K. Shanmukha Krishna**
20ME1A0593	**20ME1A05A6**	**20ME1A0569**	**20ME1A0596**

ABSTRACT

Agricultural productivity hinges on a myriad of factors, each playing a vital role in shaping crop growth and yield. This abstract offers an insight into the significance of key factors such as nitrogen, phosphorus, potassium, pH, temperature, humidity, rainfall, and type of soil in agricultural practices. Nitrogen, phosphorus, and potassium are the indispensable nutrients critical for plant metabolic functions, growth, and productivity. Their availability in soil profoundly impacts crop health, necessitating precise fertilization strategies. Soil pH, governing nutrient uptake and microbial activity, underscores the importance of maintaining optimal levels for crop development. Temperature dictates crucial growth stages, with crops exhibiting specific temperature requirements for germination, growth, and reproduction. Humidity influences transpiration rates, disease susceptibility, and overall plant vigor, necessitating judicious irrigation and crop management practices. Rainfall patterns profoundly influence agricultural outcomes. Adequate rainfall ensures soil moisture, sustains plant growth, and supports crop yields. Soil type, defined by texture, structure, and composition, shapes water retention, nutrient availability, and root growth. Tailored soil management, including amendments and rotations, optimizes fertility and productivity. Using Random Forest Classifier model is build based on the features to predict the suitable crop and crop yield.

The impact of climate change in India, most of the agriculture crops are badly affected in term of their performance over a period of last decades. Predicting the crop yield in advance of its harvest would help the policy makers and farmers for taking appropriate measure for marketing and storage. The result of the prediction will made available to the farmer. Thus, for such kind of data analytics in crop prediction, there are different techniques or algorithm is used, and with the help of those algorithms we can predict the crop yield. The main concept is to increase the throughput of the agriculture sector with the machine learning models. Machine learning model are trained using the datasets, and the expected result are based on previous experience. The datasets include the information of past years weather conditions like rainfall, temperature etc. We can determine the parameter for the model during the training phase by evaluating past data. A portion of the previous data is considered in the evaluation. Weather prediction is the important responsibility in agriculture. However, it is the tough process to predict manually. Too many elements affect agriculture productivity since it is not dependent on a single factor. So, we decided to train the model based on weather conditions like rain, temperature etc.

CHAPTER-1

INTRODUCTION

1.1 Introduction

Crop yield prediction is an essential predictive analytics technique in the agriculture industry. It is an agricultural practice that can help farmers and farming businesses predict crop yield in a particular season when to plant a crop, and when to harvest for better crop yield. Predictive analytics is a powerful tool that can help to improve decision-making in the agriculture industry. It can be used for crop yield prediction, risk mitigation, reducing the cost of fertilizers, etc. The crop yield prediction using ML and flask deployment will find analysis on weather conditions, soil quality, fruit set, fruit mass, etc.

Agriculture plays a vital role in Indian economy. Our Project helps farmers to get more yields of crops can be achieved by analyzing agro-climate data using machine learning techniques. Machine Learning is emerging research field in crop yield analysis. Yield prediction is a very important issue in agricultural. Any farmer is interested in knowing how much yield he is about to expect. In the past, yield prediction was performed by considering farmer's experience on particular field and crop. The yield prediction is a major issue that remains to be solved based on available data.

Machine learning model are trained using the datasets, and the expected result are based on previous experience. We can determine the parameter for the model during the training phase by evaluating past data. A portion of the previous data is considered in the evaluation.

Weather prediction is the important responsibility in agriculture; however, it is the tough process to predict manually. Too many elements affect agriculture productivity since it is not dependent on a single factor. The logistic regression is the simple supervised machine learning algorithm which train the dataset and that dataset will predict the suitable crop for the yield. Thus, the main objective of our paper is to predict the suitable crop based on weather condition.

1.2 Overview of the Project

Our project represents a groundbreaking initiative focused on revolutionizing agricultural management practices. By harnessing the power of predictive analytics and machine learning techniques, we aim to enhance the efficiency and productivity of farming operations. Through the integration of advanced data analysis methods, we seek to provide farmers and agricultural stakeholders with valuable insights for optimizing crop yield prediction and resource allocation strategies.

At the core of our project lies the ambition to develop robust predictive models capable of accurately forecasting crop yields. By leveraging historical and real-time data on environmental conditions, soil characteristics, and crop phenology, we strive to uncover hidden patterns and relationships that influence crop growth and development. Through collaboration with experts in the field and the utilization of cutting-edge technologies, we aspire to empower stakeholders with actionable insights to make informed decisions and drive sustainable agricultural practices forward.

The Crop Yield Prediction uses the various farm inputs, weather conditions, type of soil is used to predict the suitable crop in the particular scenario and the predicted crop yield in terms of kilograms per hectare. The Farm inputs include the seeds, fertilizers, pesticides, machinery, and labour. In our project we are considering only the fertility of the soil that is the ability of soil to sustain plant growth and optimize crop yield.

There are two types of nutrients in the soil are macro and micro nutrients. Macronutrients are Nitrogen (N), Phosphorus (P), Potassium (K) and Micronutrients are Manganese (Mn), Boron (B), Copper (Cu), Iron (Fe), Zinc (Zn), Nickel (Ni), Molybdenum (Mo) and Chlorine (Cl). The availability of N, P and K in soil should be sufficient, but not too high. Too low availabilities will lead to hampered growth and low yields, while too high availabilities of one or more nutrients may lead to disturbed plant growth and adverse effects for yield and/or quality of harvested products.

Moreover, the N, P and K availability should be balanced, so the availability of the other nutrients should be taken into account while the availability of the considered nutrient is adjusted. It is important for farmers to know the NPK content in their soil. For the optimal growth of crops, sufficient amounts of nutrients should be available in the root zone of the crops.

1.3 Objective

In crop yield prediction based on weather conditions, the primary objectives are to Improve Agricultural Planning which provide farmers with reliable forecasts of crop yields based on weather conditions to assist in planning planting schedules, resource allocation, and crop management practices. Optimize Resource Allocation are enable efficient allocation of resources such as water, fertilizers, and pesticides by predicting crop yields accurately and adjusting inputs based on anticipated weather conditions.

Improve overall agricultural productivity and sustainability by optimizing crop yields, reducing input costs, minimizing environmental impacts, and enhancing resilience to climate-related challenges. Analysing Market and Forecast Assist market analysts, traders, and policymakers in forecasting supply and demand dynamics, price fluctuations, and market trends based on projected crop yields and weather forecasts.

1.4 Scope

The scope of a Crop Yield Prediction project based on weather conditions can vary depending on factors such as the available data, resources, and specific objectives. Some of the key components that typically fall within the scope of such as

- **Support Food Security:** Contribute to food security and resilience by ensuring stable and predictable crop production, reducing vulnerabilities to weather-related disruptions, and enhancing the reliability of food supply chains.

- **Enable Research and Innovation:** Support research efforts in agronomy, climate science, and data analytics to develop advanced models, tools, and technologies for crop yield prediction, precision agriculture, and climate-smart farming practices. Proper

- **Agricultural Planning:** Provide farmers with proper forecasts of crop yields based on weather conditions to assist in planning planting schedules, resource allocation, and crop management practices.

- **Visualization and Interpretation:** Visualizing historical weather patterns, crop yields, and model predictions using charts, graphs, and maps.

1.5 Expected Outcome

The expected output of a crop yield prediction based on weather conditions are typically includes:

Predicted Crop Yields: The main output is the forecasted yield of crops for a specific time period (e.g., growing season) and geographic area (e.g., farm, region). This prediction is usually provided in quantitative terms, such as expected bushels per acre or tons per hectare.

Weather Conditions and Forecasts: The output may include information about historical weather conditions, current weather data, and forecasted weather patterns relevant to crop growth and development. This could include variables such as temperature, rainfall, humidity, wind speed.

Monitoring and Maintenance: Monitoring model performance and data quality over time, with periodic updates and recalibration as needed. Incorporating feedback from users and stakeholders to improve model accuracy, usability, and relevance.

CHAPTER-2

LITERATURE SURVEY

The Crop Yield Prediction uses the various farm inputs, weather conditions, type of soil is used to predict the suitable crop in the particular scenario and the predicted crop yield in terms of kilograms per hectare.

The Farm inputs include the seeds, fertilizers, pesticides, machinery, and labour. In our project we are considering only the fertility of the soil that is the ability of soil to sustain plant growth and optimize crop yield. There are two types of nutrients in the soil are macro and micro nutrients. Macronutrients are nitrogen (N), phosphorus (P), potassium (K) and Micronutrients are Manganese (Mn), Boron (B), Copper (Cu), Iron (Fe), Zinc (Zn), Nickel (Ni), Molybdenum (Mo) and Chlorine (Cl). The availability of N, P and K in soil should be sufficient, but not too high. Too low availabilities will lead to hampered growth and low yields, while too high availabilities of one or more nutrients may lead to disturbed plant growth and adverse effects for yield and/or quality of harvested products. Moreover, the N, P and K availability should be balanced, so the availability of the other nutrients should be taken into account while the availability of the considered nutrient is adjusted.

It is important for farmers to know the NPK content in their soil. For the optimal growth of crops, sufficient amounts of nutrients should be available in the root zone of the crops. Those nutrients can be partly supplied by the soil and should be partly added with organic manures and fertilizers. Soils will contain different amounts of available nutrients, depending of the parent material (e.g. sand, clay, peat), and differences in the management history such as preceding crops, management of crop residues and use of manure and fertilizers in the past. Also differences in climatic conditions may alter the available nutrients. For that reason, it is of importance for farmers to know the NPK content of their soil, so that they know how much N, P and K they should add with organic or mineral fertilizers, to optimize crop growth, production and yield.

Weather Conditions include the Temperature, Humidity, rainfall. Temperature affects the growth and development of crops and influences the timing of planting, flowering, and harvesting. Different crops have specific temperature requirements for germination, growth, and maturity. Frost and extreme cold can damage or kill sensitive plants, while prolonged periods of high temperatures can stress crops and reduce yields. Humidity levels affect transpiration rates in plants, which is the process of water vapor escaping from plant leaves. Proper humidity is important for maintaining balanced water levels in plants. High humidity can encourage the development of fungal diseases, while low humidity can lead to excessive water loss from plants. Rainfall can all affect the amount of water available for irrigation.

Soil can have the characteristics of their colour, texture and chemical properties. There are different types of soil. They are as follows - Soil Classification on the based on colour are Black Soil, Red Soil, and Soil Classification on the Based-on Particles Size are Sandy Soil, Clayey Soil, Loamy Soil. Red soil is suitable for growing groundnuts, pulses, millet, cotton and tobacco. Black soil is ideal for growing crops such as paddy, sugarcane, wheat. Sandy soil is not good for plants. However, melon and coconut grow in sandy soil. Clay soil is only good for crops like paddy, which require a lot of water. Clay is used for making toys, pots, and many other purposes. Loamy soil is ideal for growing Vegetables also grow well in this soil.

1. "Machine learning approach for forecasting crop yield based on climatic parameters"

Author: S.Veenadhari, Dr. Bharat Misra, Dr. CD Singh

Climate plays an important role in the field of agriculture. Over this year due to increase in global warming climate has been affected badly and it had a great impact on crops. Predicting the crop yield will tell the farmers what to harvest depending upon the predictive analysis.

2. "Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique"

Author: Rakesh Kumar1, M.P. Singh2, Prabhat Kumar 3 and J.P. Singh

Food Security and Economic Growth are one of the important factors in the field of Agriculture across the agro-based countries. Crop Selection is a difficult task for agriculture planning depending on the climate. It depends on various aspects such as climatic conditions, Market price, production rate and Government policy.

3."Agro consultant: intelligent crop recommendation system using machine learning algorithms"

Author: Zeel Doshi, Subhash Nadkarni Agriculture is the main backbone of the Indian economy. The Indian population depends either internally or externally on agriculture for their livelihood. Thus, agriculture plays a key role in the country. Many of the farmers believe in harvesting crops by considering some major factors like guessing, seasonal factors and their previous experience.

4.“Title: A Survey on Data Mining Techniques in Agriculture, 2015”

Author: M. C. S. Geetha, Horticulture is important for the economies of emerging nations, particularly India. Mining assumes a significant part in decision-production in numerous spaces of farming. It inspects the job of data mining in the farming area and work corresponding to a couple of creators in the rural area. It likewise checks out various methods of acquiring data to resolve numerous agrarian issues. 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.

CHAPTER-3

SYSTEM ANALYSIS

3.1 Existing System

There are many machine learning models related to the crops. The Existed system contains the inputs like the type of soil, weather conditions like temperature, humidity, rainfall. The output will be only the recommended crop.

The other models related to the crop yield for the particular crop that takes the inputs are the type of soil, weather conditions like temperature, humidity, rainfall and type of crop and produce the crop yield at the end of the season in terms of the kilograms per hectare.

So, the existed model is not effectively can help the farmers because the model only predicts the output of the crop. It will not recommend the suitable crop for those conditions.

Disadvantages

- **Limited Predictive Accuracy:** The existing system relies on basic statistical methods, lacking advanced predictive analytics capabilities, resulting in less accurate crop yield forecasts.
- **Manual Data Processing:** Data collection and preprocessing are primarily manual, leading to inefficiencies and potential inaccuracies.
- **Limited Scalability:** The system struggles to handle large datasets or adapt to evolving agricultural conditions and requirements.
- **Lack of Real-time Insights:** Without real-time data processing capabilities, the system cannot provide timely insights for informed decision-making.

3.2 Proposed System

The Proposed System came against the disadvantage of the existed system. The proposed system contains the inputs are the type of soil, weather conditions like temperature, humidity, rainfall and also most important factors in the soil are N, P and K. The output will be recommended crop and the crop yield in the terms of the kilograms per hectare.

Crop yield prediction is an important aspect of agriculture that helps farmers make informed decisions about their crops. It involves estimating the number of crops that will be produced in a given area based on various factors such as soil type, weather conditions, and crop management practices. In recent years, machine learning (ML) has emerged as a powerful tool for predicting crop yields.

To implement machine learning for crop yield prediction, a large dataset of crop yield data is required. This data should include information about the crop, such as the type of crop, the location, and the date of planting. Additionally, data on weather conditions and soil characteristics should also be collected. The machine learning algorithm is then trained on this data to learn the relationships between the inputs and outputs

Advantages

- **Enhanced Predictive Accuracy:** Utilizing advanced machine learning algorithms like Random Forest, the proposed system offers improved predictive accuracy and reliability in forecasting crop yields.
- **Automated Data Processing:** The system automates data collection, preprocessing, and analysis, reducing manual intervention and minimizing errors.
- **Scalability:** Designed with scalability in mind, the system can efficiently handle large datasets and adapt to changing agricultural landscapes and demands.
- **Real-time Insights:** With real-time data processing capabilities, the proposed system delivers timely and actionable insights, empowering stakeholders to make informed decisions promptly.

CHAPTER-4

SYSTEM STUDY

4.1 Feasibility Study

Crop yield prediction is an essential predictive analytics technique in the agriculture industry. It is an agricultural practice that can help farmers and farming businesses predict crop yield and predict the suitable crop in a particular season. The machine learning model will take the input from the user. The main challenge in the project was the predicting the weather conditions in advance.

4.1.1 Operational Feasibility

The machine learning model concerns with the past data that will definitely solve the problem of predict the crop yield and suitable crop for the input conditions. The model contains the previous year data that can used to predict the correct values for the input values. The model operates according to the N, P and K, type of soil and weather conditions it will predict the output.

Leveraging machine learning algorithms and robust data integration techniques, the system processes diverse datasets to generate right predictions and actionable insights. Moreover, the incorporation of soil nutrient analysis, particularly NPK content, enhances the predictive capabilities, allowing for tailored recommendations based on soil health. The user interface is designed with simplicity and usability in mind, offering farmers intuitive access to input parameters and visualization of prediction results.

4.1.2 Economic Feasibility

This model is carried out to without any economic impact on any person. The amount of fund that the test for nutrients of soil can pour into the research and development of the system is limited. The weather data can be available without any cost. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the test for the nutrients in the soil can be expensed.

The economic feasibility of implementing a crop yield prediction using machine learning (ML) in a small organization involves a careful analysis of costs and potential benefits. While the upfront investment in hardware, software, and training may pose challenges for smaller budgets, the long-term advantages can outweigh these costs

4.1.3 Technical Feasibility

The technical feasibility study is carried out to check the technical feasibility, that is, the technical requirements of the system. The system developed must not have a high demand on the available technical resources. The technical resources are IDE - PyCharm, Python can be used. It will not lead to high demands being placed on the users and developers.

PyCharm is a popular integrated development environment (IDE) specifically designed for Python programming. It is developed by JetBrains and provides a comprehensive set of tools for Python development, including code editing, debugging, syntax highlighting, code completion, version control integration, and support for various frameworks and libraries commonly used in Python development.

Google Colab, short for Google Colaboratory, is a free cloud service provided by Google that allows users to write and execute Python code in a browser-based interactive environment. It is particularly popular among data scientists, machine learning researchers, and educators for its ease of use and the ability to harness the computational power of Google's infrastructure without requiring any setup.

NumPy is a fundamental package for numerical computing with Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is widely used in scientific computing, data analysis, machine learning, and various other domains where numerical operations on large datasets are common.

Pandas is a powerful and widely used open-source data manipulation and analysis library for Python. It provides data structures and functions designed to make working with structured and tabular data easy and intuitive. Pandas is particularly popular among data scientists, analysts, and researchers for tasks such as data cleaning, exploration, transformation, and analysis.

4.2 System Requirements

4.2.1 Hardware Requirements

- **Processor:** A modern multicore processor, such as an Intel Core i5 or i7, is recommended for training and deploying machine learning models.
- **RAM:** 4GB and Higher.
- **Hard Disk:** 500GB Minimum.
- **Fast storage:** A fast storage device, such as an SSD, is needed for storing and loading datasets and models.

4.2.2 Software Requirements

- **Operating System:** Most machine learning frameworks are compatible with popular operating systems such as Windows, macOS, and Linux.
- **Python:** Python is the most commonly used programming language for machine learning, and most machine learning frameworks are built using Python.
- **Frame Works:** There are many machine learning frameworks available, such as TensorFlow, Keras, and scikit-learn. These frameworks provide tools and APIs for building, training, and deploying machine learning models.
- **Development Tools:** IDEs such as Google Colab and PyCharm are commonly used for machine learning development.
- **Libraries:** Libraries such as NumPy, Pandas, and Matplotlib are commonly used for data manipulation, analysis, and visualization.

CHAPTER-5

PROJECT ARCHITECTURE

5.1 System Architecture:

System architecture is represented in below diagram. The machine learning model starts from the dataset containing the required features, later the Preprocessing data contains techniques like handling missing data, data cleaning. The exploring data is a crucial step in the machine learning pipeline that involves correlation analysis and visualizing the dataset to gain insights, understand the underlying patterns, and inform subsequent modelling decisions. The train-test split is a fundamental step used to evaluate the performance of a model. It involves dividing the dataset into two subsets: one for training the model and the other for testing its performance. This split allows us to assess how well the trained model generalizes to unseen data. we can perform a train-test split using the scikit-learn library in python. The train and evaluating process allows you to train and evaluate machine learning algorithms on your dataset, helping you understand how well the model performs on unseen data and guiding you in the selection of the best-performing algorithm for your task. Using Flask with machine learning allows you to create web applications that serve machine learning models.

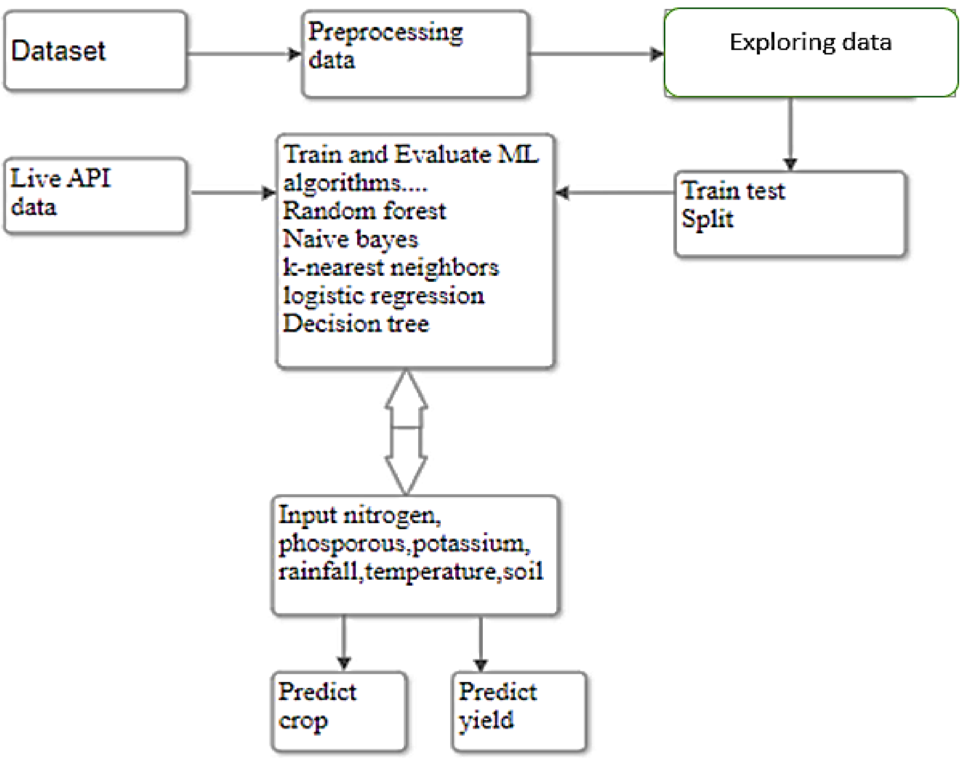


Fig: System Architecture for Crop yield prediction

5.2 Design and Diagrams

In the design and diagrams section, we will present visual representations of the system architecture and components. This includes the following diagrams:

Use Case Diagram: This diagram illustrates the interactions between users (actors) and the system, depicting the various functionalities and features available to different user roles.

Class Diagram: The class diagram represents the static structure of the system, depicting the classes, their attributes, and the relationships between them.

Sequence Diagram: The sequence diagram visualizes the dynamic behavior of the system by illustrating the sequence of interactions between objects or components over time.

Collaboration Diagram: This diagram, also known as a communication diagram, complements the sequence diagram by showing the interactions between objects or components in a more intuitive and visual manner.

Data Flow Diagram: A data flow diagram (DFD) is a graphical representation of the flow of information through a system. It shows the processes that transform data, the data stores that hold data, and the external entities that interact with the system. DFDs are used in system design and analysis to document how a system works and to identify areas for improvement.

5.2.1 Use case Diagram

1.Actor: An actor represents a role that interacts with the system. Actors can be human users, external systems, or even other software components. In a use case diagram, actors are typically depicted as stick figures.

Actors are: Developer, End user

2.Use Case: A use case represents a specific functionality or a task that the system performs. It describes a sequence of actions that the system performs, yielding an observable result of value to a particular actor. Use cases are represented by ovals in the diagram.

Use cases are: collect and load dataset, preprocessing data, Exploring data, create model, fit model, performance metrics, export model, deploy model, Access server.

3.Association: An association is a relationship between an actor and a use case. It indicates that an actor participates in a particular use case. Associations are represented by lines connecting actors to use cases.

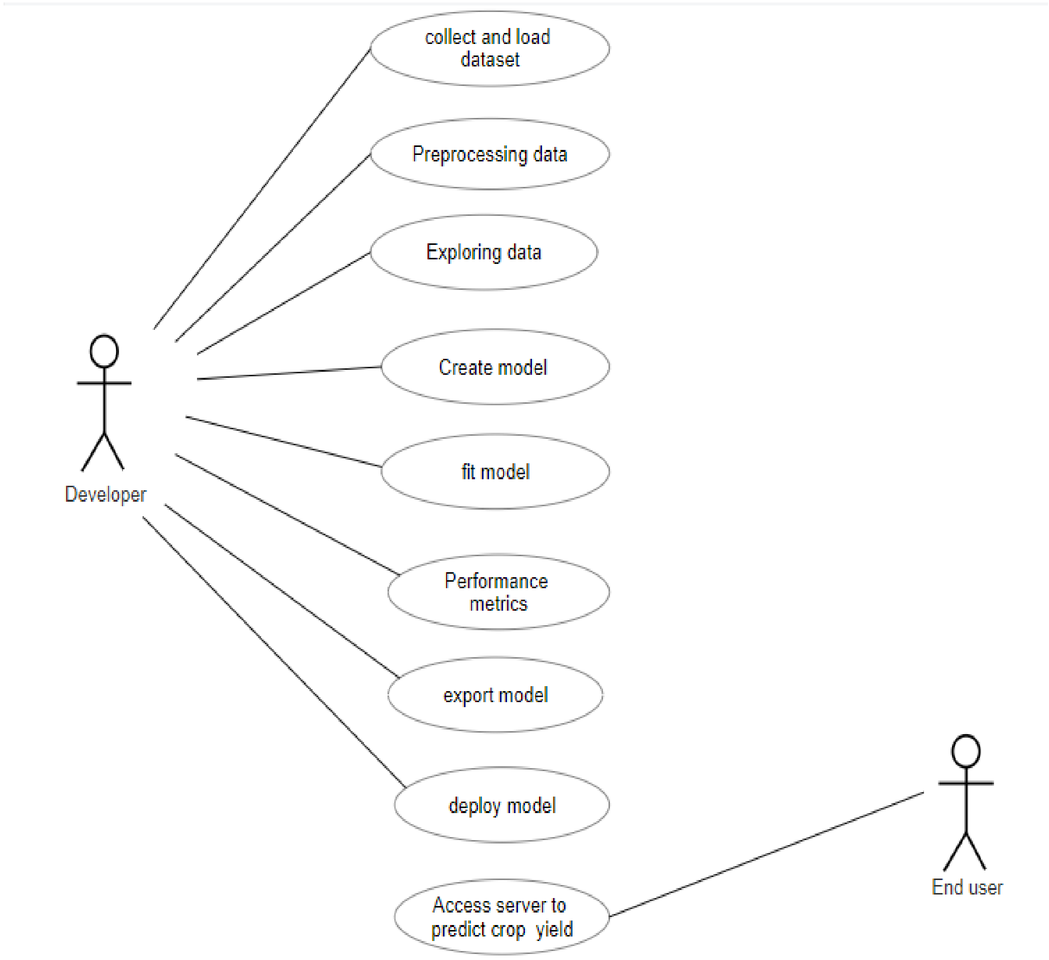


Fig: Use case Diagram for Crop yield prediction

5.2.2 Class Diagram

System: This is the overarching class that encapsulates the entire crop yield prediction system. It likely has the functionality to manage the overall process and coordinate the activities of the other classes.

User: This class represents the system's user, typically a farmer or agricultural specialist. It provides methods for the user to interact with the system, such as uploading datasets.

Dataset: This class represents the agricultural data used to train and test the machine learning models for yield prediction. It likely has attributes to store the data itself and methods to access and manipulate it.

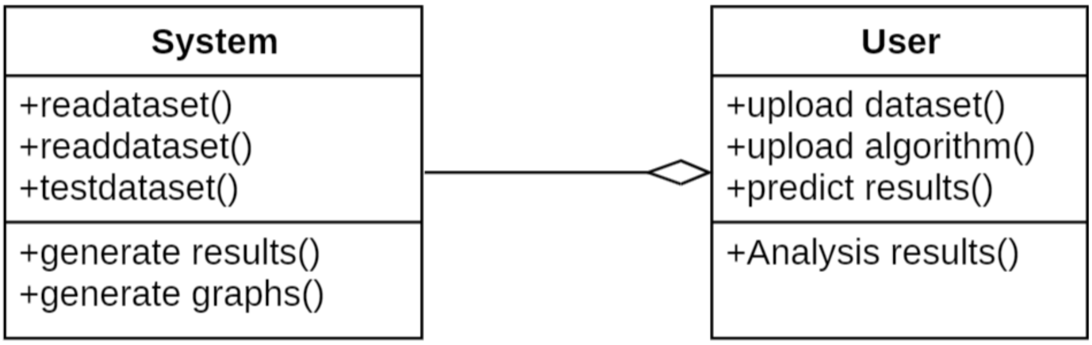


Fig: Class Diagram for Crop yield prediction

5.2.3 Sequence Diagram

The sequence diagram you sent illustrates the process of collecting data, training a model, and using the model to predict crop yield. Here's a breakdown to include in your project report:

Actors: The diagram involves two actors:

Developer/End User (Farmers/Agricultural Advisors): This actor represents the person initiating the process, such as a data scientist or agricultural specialist.

Server: This actor represents the system that performs the data processing, model training, and prediction tasks.

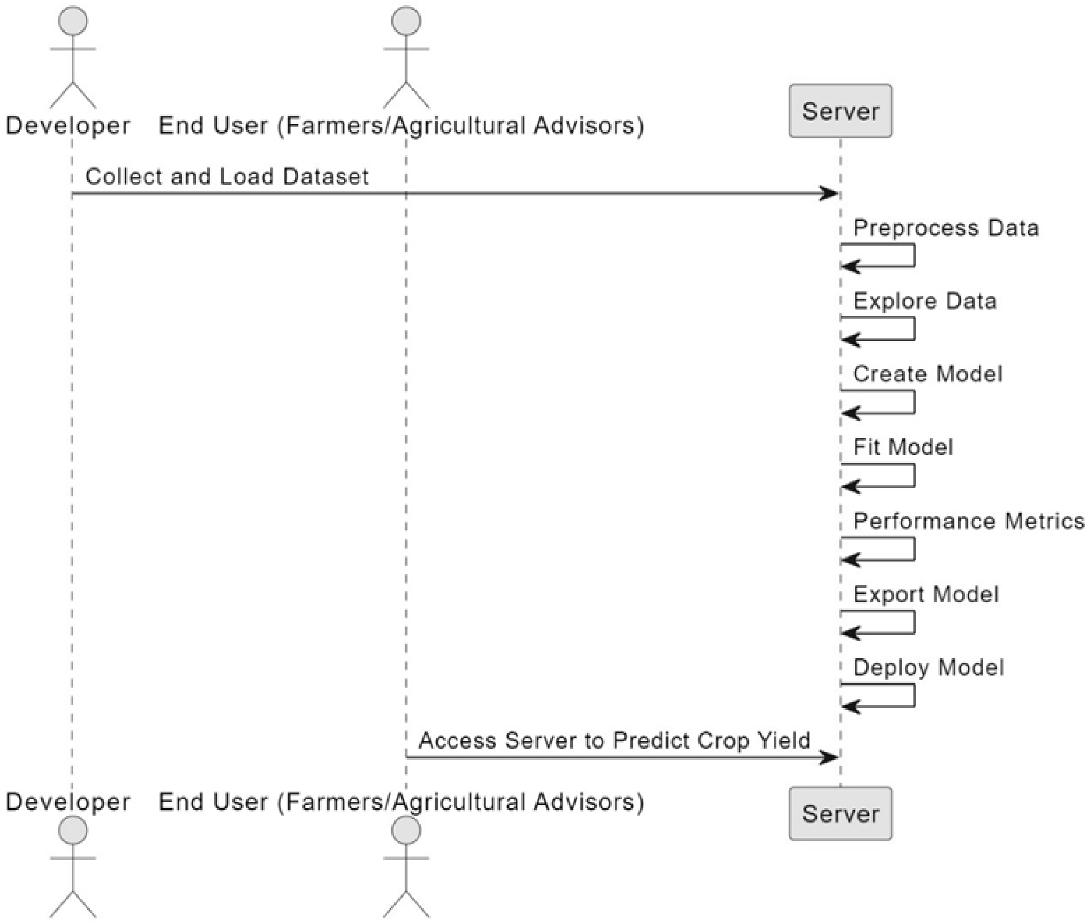


Fig: Sequence Diagram for Crop yield prediction

5.2.4 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, offers a different perspective compared to a data flow diagram. It focuses on how objects interact with each other to achieve a specific task, rather than the flow of data itself.

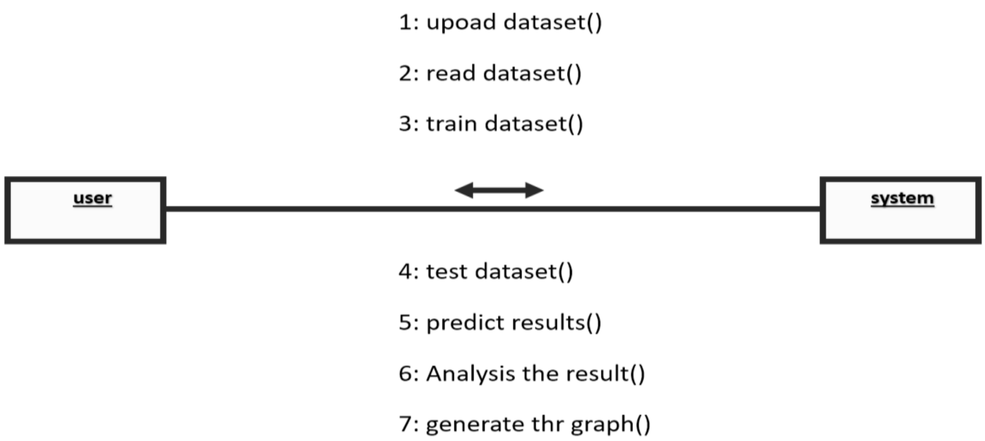


Fig: Collaboration Diagram for Crop yield prediction

5.2.5 Data Flow Diagram

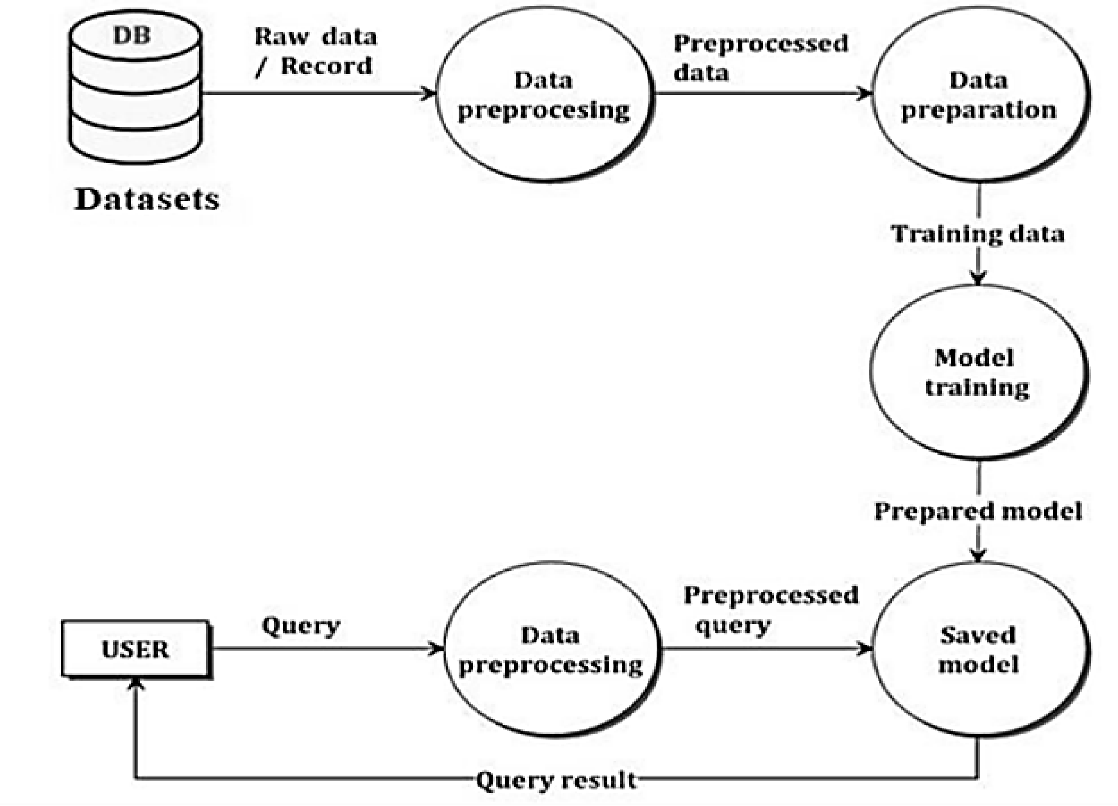


Fig: Level – 1 Data Flow Diagram for Crop yield prediction

Stage 1: Data Preparation and Model Training

- Raw agricultural data is collected (weather, soil, yield etc.)
- This data is preprocessed and cleaned for use in models.
- The prepared data is split into training and testing sets.
- The training data is used to train a machine learning model (e.g., Random Forest Regression).
- A trained model is generated, ready for prediction.

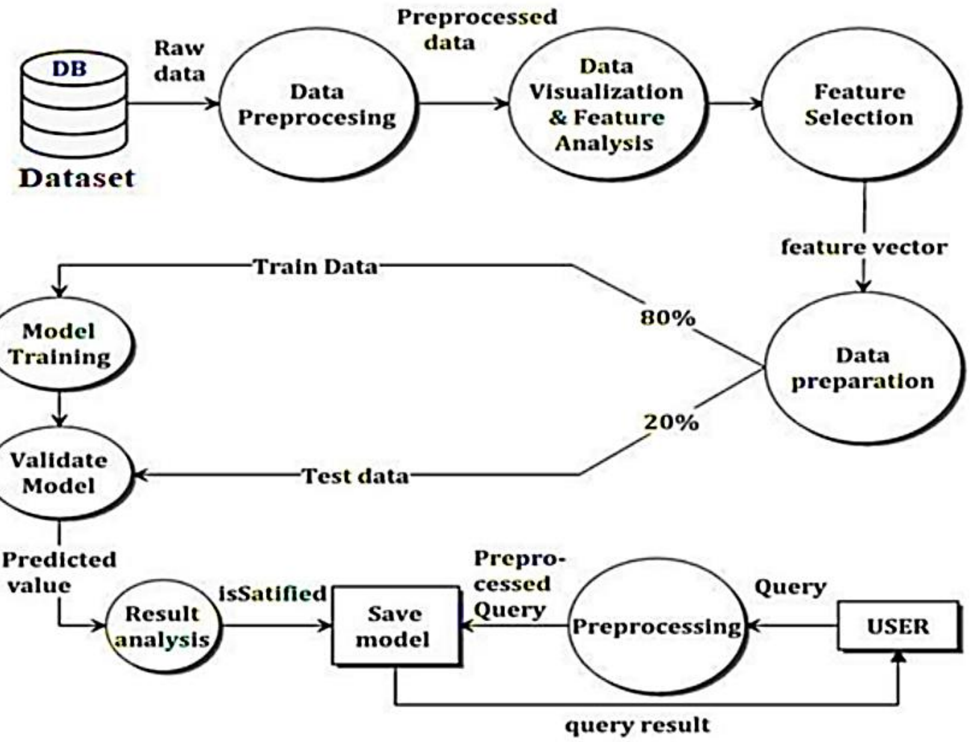


Fig: Level - 2 Data Flow Diagram for Crop yield prediction

Stage 2: Model Usage and Result Analysis

- Users provide new data (specific field conditions) for prediction.
- This new data is preprocessed similarly to the training data.
- The trained model is used to predict crop yield for the user's data.
- The predicted yield is presented to the user.

Overall Data Flow

The Level 1 processes prepare the data and train the model, while Level 2 utilizes the trained model to make predictions on new data and present the results to the user. The data flows from raw data sources through preprocessing, training, and prediction stages to generate a crop yield prediction for the user.

CHAPTER-6
METHODOLOGY

6.1 Data Dictionary

- **Temperature:** Temperature is measured in degrees Celsius and represents the thermal condition of the environment in which crops are growing. Temperature influences various biological processes in plants, including photosynthesis, respiration, and flowering, making it a critical factor in determining crop growth and development.

- **Humidity:** Humidity refers to the amount of water vapor present in the air and is expressed as a percentage of the maximum amount of water vapor the air can hold at a given temperature. High humidity levels can promote the spread of fungal diseases in crops, while low humidity levels can increase water stress in plants, affecting their growth and productivity.

- **Rainfall:** Rainfall denotes the amount of precipitation, typically measured in millimeters, that falls over a specific area during a certain period. Adequate rainfall is essential for supplying water to crops, supporting their growth and development, while excessive or insufficient rainfall can lead to waterlogging or drought stress, respectively, affecting crop yields.

- **Nitrogen Level:** Nitrogen level in the soil is quantified in parts per million (ppm) and indicates the availability of nitrogen, a key nutrient required for plant growth and protein synthesis. Nitrogen deficiency can result in stunted growth and reduced yield, while excess nitrogen can lead to environmental pollution and nutrient imbalances in crops.

- **Phosphorus Level:** Phosphorus level in the soil is measured in parts per million (ppm) and represents the concentration of phosphorus, an essential nutrient involved in energy transfer, photosynthesis, and root development. Adequate phosphorus levels are critical for promoting root growth, flowering, and fruit formation in plants, contributing to overall crop productivity and quality.

- **Potassium Level:** Potassium level in the soil is measured in parts per million (ppm) and indicates the availability of potassium, an essential nutrient for plant growth.

Potassium plays a crucial role in various physiological processes within plants, including enzyme activation, osmoregulation, and stress tolerance, making it essential for overall crop health and yield optimization.

- **Soil:** Soil fertility is one of the most important factors in crop production, as it's the soil's ability to provide nutrients for a crop to grow optimally. Soil quality, such as its pH, texture, structure, and nutrient content, can have a significant impact on plant growth and yield. yield, while low soil moisture can cause moisture stress that affects crop productivity

6.2 Data Collection

The Data Collection contains details the specific data required, sources for acquiring the data, and considerations for ensuring data quality.

1. Data Requirements

The project aims to develop a machine learning model for predicting crop yields. To achieve this, we will collect the following data for achieving high quality Historical Crop Yield Data:

- **Crop type-** It is the type of crop that is cultivated. (e.g., rice, maize)

- **Yield amount-** It describes how much yield is produced for that crop per hectare. (e.g., kg per hectare)

- **Soil type-** Based on the nature of soil which crop is suitable for better yield.

Weather Data:

- **Temperature-** It is the average temperature in cultivation of that crop (daily average, minimum, maximum)

- **Rainfall-** it is amount of rainfall in mm for that crop required (total amount, frequency)

- **Humidity-** The level of humidity suitable for specific crop cultivation. (average relative humidity)

- **pH level-** In soil pH consideration is most important for specific crops.

- **Nutrient content-** Soil nutrients is required in different quantity for different crops, mostly nutrients include nitrogen, phosphorous and potassium. (nitrogen, phosphorus, potassium).

2. Data Sources

We have explored various sources to collect the required data, aiming for a balance between comprehensiveness and accessibility. Collection of data is of two types of firstly Primary data which is collected by us in various ways, and another is secondary data which is easy to gather and analyse it accordingly to our data requirement. Most of the data collected for our model is Secondary data which is mostly focused on qualitative approach of data. Here's a breakdown of potential sources:

- Government Agencies: Ministry of Agriculture (or relevant department) in our country “https://agriwelfare.gov.in/”
- Other government agencies responsible for agriculture and weather data Agricultural Research

Institutions:

Websites of research institutions focusing on agriculture and crop science

- ICAR-National Bureau of Plant Genetics Resources: New Delhi
- ICAR-National Bureau of Soil Survey and Land Use Planning: Nagpur
- Indian Institute of Soil Sciences: Bhopal
- Central Research Institute of Dryland Agriculture: Hyderabad
- National Rice Research Institute: Cuttack Publications and datasets available through these institutions. But the most of the data is gathered in Kaggle and GitHub.

6.3 Preprocessing Techniques

The data preprocessing phase is the most challenging and time-consuming part in the machine learning project, but it's also one of the most important parts. If you fail to clean and prepare the data, it could compromise the model.

When dealing with real-world data, we always need to apply some preprocessing techniques in order to make the data more usable. These techniques will facilitate its use in machine learning (ML) algorithms, reduce the complexity to prevent overfitting, and result in a better model. The techniques that we'll explore are:

- **Data Cleaning**
- **Data Transformation**

Data Cleaning

One of the most important aspects of the data preprocessing phase is detecting and fixing bad and inaccurate observations from your dataset in order to improve its quality. This technique refers to identifying incomplete, inaccurate, duplicated, irrelevant or null values in the data. After identifying these issues, you will need to either modify or delete them.

Data Transformation

Converting the Data to the Same Structure

1. **Transformation for categorical variables:**

Categorical variables, usually expressed through text, are not directly used in most machine learning models, so it's necessary to obtain numerical encodings for categorical features. The approach you use will depend on the type of variables.

2. **Minmax scaler/ normalization:**

The min-max scaler, also known as normalization, is one of the most common scalers and it refers to scaling the data between a predefined range (usually between 0 and 1). The main issue with this technique is that it's sensitive to outliers, but it's worth using when the data doesn't follow a normal distribution.

3. **Standard scaler:**

The Standard Scalar is another widely used technique known as z-score normalization or standardization. It transforms the data so that the mean of the data is zero and the standard deviation is one. This approach works better with data that follows the normal distribution and it’s not sensitive to outliers.

6.4 Prepared Data

Prepared data, is known as data after preprocessing, it is the process of gathering, organizing, and transforming raw data to make it suitable for analysis. This process includes cleaning, normalizing, and consolidating data, as well as adding new dimensions or attributes. Data preparation is a critical part of machine learning, and the quality of the analysis depends on the quality of the input data.

Data preparation is the first step in data analytics projects. The steps include: Gathering data, Discovering and assessing data, Cleaning and validating data, Transforming and enriching data, and Storing data.

Data cleaning is the process of removing faulty data and filling in gaps. This involves:

- Checking for duplicate or invalid records
- Deleting or correcting them
- Filling in the missing values using methods such as mean, median, mode, or regression.

Data transformation is the process of updating the format or value entries in order to reach a well-defined outcome, or to make the data more easily understood by a wider audience.

CHAPTER-7

MODEL

In the Crop Yield Prediction System, the training and validation process is crucial for developing accurate predictive models. The process begins with the collection of historical agricultural data from various sources, including weather stations, agricultural databases, and government agencies. This data encompasses a wide range of variables, such as weather conditions (temperature, humidity, rainfall), soil fertility indicators (nitrogen, phosphorus, potassium levels), and crop yields.

Once the data is collected, it undergoes preprocessing to ensure its quality and suitability for model training. Preprocessing tasks include handling missing values, scaling numerical features, and encoding categorical variables. Exploratory data analysis (EDA) techniques are then applied to gain insights into the relationships between different variables and identify potential predictors of crop yields.

Next, suitable machine learning algorithms are selected for building predictive models. Decision trees, random forests, and gradient boosting machines are commonly employed due to their ability to capture complex relationships in the data. The selected models are trained using the preprocessed data, with the training process involving feeding the models with input data and adjusting their parameters iteratively to minimize prediction errors.

Validation of the trained models is performed using techniques such as holdout validation or cross-validation. A portion of the dataset is held out as a validation set, and the models are evaluated on this set to ensure their generalization ability and detect overfitting. Performance metrics such as accuracy, precision, recall, and F1 score are calculated to evaluate the effectiveness of the trained models.

In the testing and evaluation phase of the Crop Yield Prediction System, the focus shifts to assessing the performance of the trained models in real-world scenarios. This phase is critical as it determines the practical applicability and reliability of the predictive models developed during the training phase. The primary objective here is to validate the efficacy of the models in making accurate predictions on unseen data, simulating the conditions that the models will encounter when deployed in production environments.

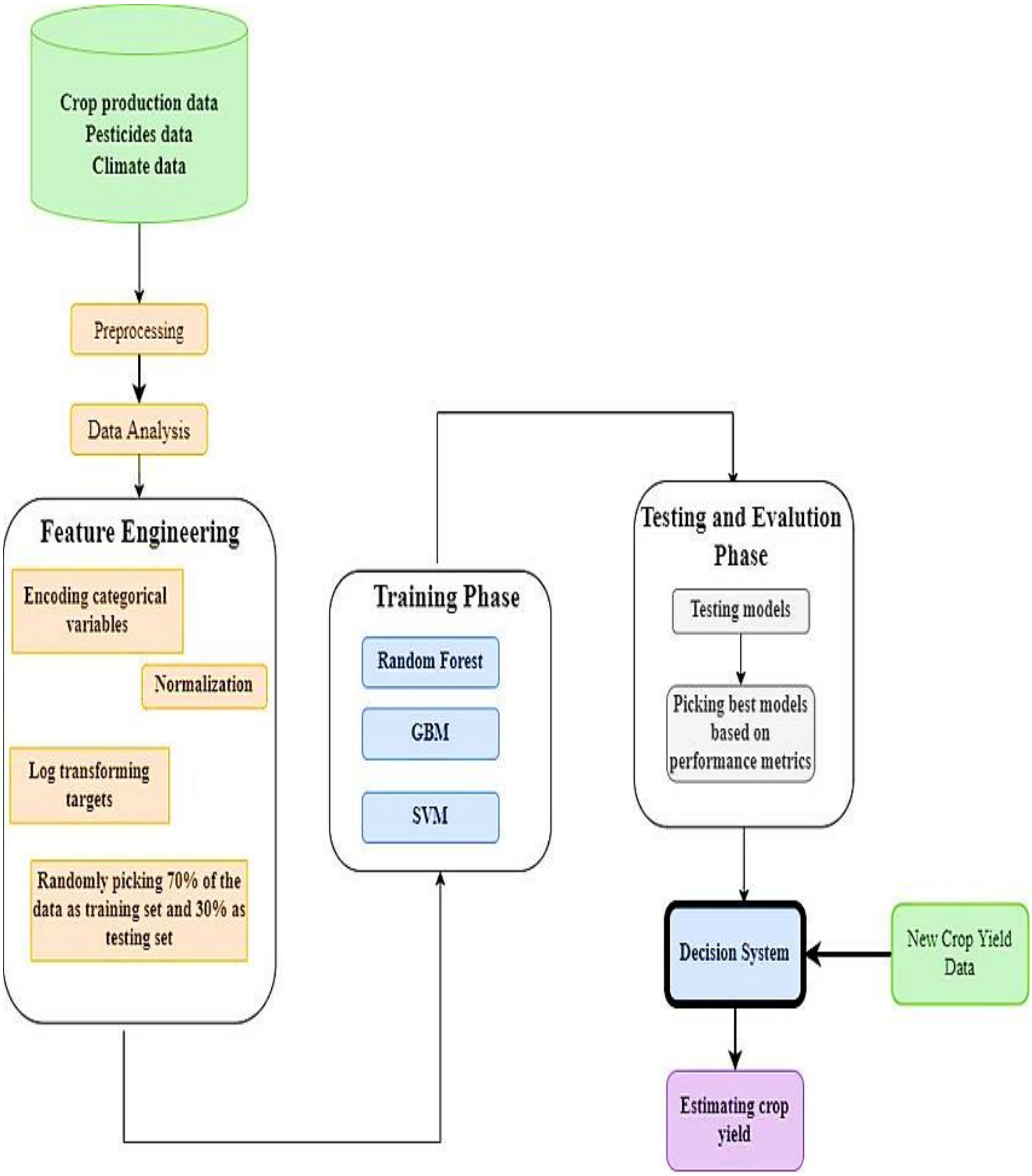


Fig: Model Diagram for Crop yield prediction

The trained models, which have successfully passed validation during the training phase, are then deployed to a server environment, making them accessible to end users for real-time crop yield prediction. End users, including farmers, agricultural advisors, and policymakers, interact with the deployed models through user-friendly interfaces, providing inputs such as current weather conditions and soil fertility indicators.

Upon receiving input data, the deployed models generate predictions for crop yields based on their learned patterns and relationships derived from the training data. These predictions are compared against ground truth values from the test dataset to assess the accuracy and reliability of the models predictions.

7.1 Training

- The testing data subset comprises a portion of the collected dataset, segregated from the training and validation sets.
- Its primary function is to provide an independent sample for assessing the performance of trained machine learning models.
- During evaluation, the models' predictions are juxtaposed with actual observations from the testing data to gauge their efficacy in real-world scenarios.
- Testing data plays a crucial role in validating the models' generalization capabilities and ensuring their robustness beyond the training data.
- When evaluating the performance of our predictive models, we compare the forecasted crop yields generated by the models with the actual yields recorded in the testing dataset.
- If the model predicts a crop yield of 50 tons per hectare for a particular region based on the weather and soil conditions, and the actual yield recorded in the testing data is 48 tons per hectare, this indicates a prediction error of 2 tons per hectare.
- By aggregating such prediction errors across multiple data points in the testing dataset, we can compute evaluation metrics such as mean absolute error or root mean square error to quantify the overall performance of the models.

7.2 Validation

- The dataset undergoes rigorous partitioning into training, validation, and testing subsets.
- K-fold cross-validation is applied to assess the model's performance across diverse data subsets.
- Performance metrics such as accuracy, precision, recall, and F1 score are computed to evaluate model predictions.
- Techniques are implemented to mitigate overfitting, ensuring the model's generalizability.
- Model hyperparameters are fine-tuned based on validation outcomes to optimize predictive accuracy.
- Multiple machine learning algorithms are compared to identify the most effective model.
- Prediction errors are analyzed to discern common issues and areas needing model

refinement.

- Model robustness is tested across various environmental and soil conditions to ensure consistent performance.
- Stakeholder feedback from validation results is integrated to iteratively enhance the model.
- Comprehensive documentation of validation processes and findings ensures transparency and reproducibility.

Random Forest Classifier:

Initially, a Random Forest Classifier is used to predict the type of crop to be cultivated based on the given environmental features. This is indicated by the prediction of the crop_num variable, which represents the type of crop (e.g., rice or maize). The classifier is trained to classify the crops into different categories based on the features provided.

The Random Forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is a set of decision trees (DT) from a randomly selected subset of the training set and then it collects the votes from different decision trees to decide the final prediction.

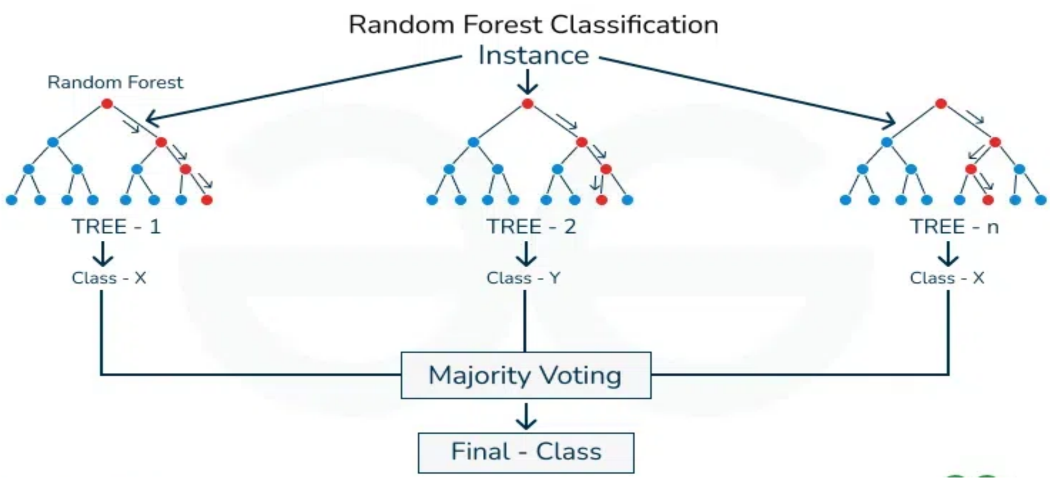


Fig: Random Forest Classification for Crop yield prediction

Decision Trees:

Decision trees are a fundamental machine learning algorithm used for both classification (predicting categories) and regression (predicting continuous values). They work by mimicking a flowchart-like structure, making them easy to visualize and understand.

Structure of a Decision Tree:

Root Node: This is the starting point, representing the entire dataset.

Internal Nodes (Decision Nodes): These nodes contain questions or conditions based on a specific feature of the data (e.g., "Is the weather sunny?"). The answer to the question determines which branch to follow.

Branches: Each branch represents a possible outcome of the test at the parent node.

Leaf Nodes (Terminal Nodes): These nodes represent the final predictions. They contain the class label (classification) or the predicted continuous value (regression).

Random Forest Regression

Random Forest Regression in machine learning is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

After determining the type of crop using the classifier, a Random Forest Regressor is used to predict the yield of the identified crop. The regressor takes the same set of environmental features but predicts a continuous variable: the yield of the specific crop identified in the classification step. The regressor is trained to estimate the yield output based on the environmental conditions for a particular crop.

Here's an expanded explanation of Random Forest Regression in the context of crop yield prediction:

Strengths of Random Forest Regression:

Reduced Overfitting: Random forests address overfitting, a common problem in machine learning where the model performs well on training data but poorly on unseen data. By combining predictions from multiple, slightly different decision trees, random forests average out these errors and lead to more generalizable results.

Handles Missing Data: Random forests can effectively handle missing data points within the training dataset. During tree construction, each node considers a random subset of features, allowing the model to bypass missing values in some features and learn from the available data.

Importance Scoring: Random forest models provide feature importance scores, which highlight the relative contribution of each environmental factor (e.g., temperature, rainfall) to the yield prediction. This helps farmers understand which factors have the most significant impact on their crops.

Implementation for Crop Yield Prediction:

1. Data Preprocessing: Prepare the data by handling missing values, scaling numerical features, and potentially performing feature engineering (creating new features based on existing ones) to improve model performance.

2. Random Forest Regression: Train a Random Forest Regression model using the environmental features (and potentially the classified crop type) as input variables and the historical crop yield data as the target variable. The model learns the relationships between these factors and yield.

3. Prediction: For a new set of environmental data (representing a specific field or location), the model predicts the expected crop yield.

Additional Considerations:

Number of Trees: The number of trees in the forest is a hyperparameter that can be tuned to optimize model performance. More trees generally lead to better accuracy but also increase computational cost.

Maximum Tree Depth: This hyperparameter controls the complexity of individual trees. Deeper trees can capture more intricate relationships but are also more prone to overfitting. Finding the right balance is crucial.

Feature Selection: Identifying the most relevant environmental features can improve model performance and interpretability. Techniques like feature importance scores can help guide this selection process.

CHAPTER-8

SOFTWARE DESCRIPTION

The software description for our project encompasses the following components:

Google Colab:

A cloud-based platform offered by Google, Google Colab provides a Jupyter notebook interface for writing and executing Python code directly within a browser-based environment. It offers free GPU and TPU support, making it ideal for accelerating computations, particularly for tasks like deep learning. Additionally, its cloud-based nature allows for easy collaboration and access from any device with an internet connection. Integration with Google Drive facilitates seamless saving, sharing, and accessing of project notebooks.

Key features of Google Colab include:

1. **Free GPU and TPU Support:** Google Colab provides complimentary access to GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units), which significantly accelerate computations, particularly for tasks like deep learning. In our project, we leverage this feature to accelerate the training of machine learning models.

2. **Cloud-Based Environment:** As a cloud-based platform, Google Colab is accessible from any device with an internet connection, facilitating collaboration and enabling users to work on projects across various devices. In our project, we utilize this feature for collaborative development and easy access to project notebooks.

3. **Integration with Google Drive:** Google Colab seamlessly integrates with Google Drive, enabling effortless saving, sharing, and accessing of notebooks directly from Google Drive accounts. This integration enhances data management and sharing capabilities within our project.

4. **Pre-installed Libraries:** Google Colab comes pre-installed with popular Python libraries such as TensorFlow, PyTorch, Keras, Pandas, and NumPy. Additionally, users have the option to install additional libraries using pip or conda. In our project, we leverage these pre-installed libraries for various data processing and machine learning tasks.

PyCharm:

Developed by JetBrains, PyCharm is an Integrated Development Environment (IDE) specifically designed for Python development. It features a robust code editor with syntax highlighting, code completion, formatting, and refactoring capabilities. PyCharm's integrated debugger aids in easy debugging of Python code, while its support for web development, including HTML, CSS, and JavaScript, makes it versatile for various development tasks. While our project primarily focuses on machine learning and data analysis, PyCharm's features enhance the development experience and streamline the coding process.

1. **Code Editor:** PyCharm boasts a robust code editor equipped with features like syntax highlighting, code completion, formatting, and refactoring capabilities. It supports multiple Python versions and provides intelligent code suggestions based on context. In our project, we utilize PyCharm's code editor for writing and editing Python scripts.

2. **Debugger:** PyCharm includes an integrated debugger, facilitating easy debugging of Python code. Developers can set breakpoints, step through code, inspect variables, and evaluate expressions to identify and resolve bugs efficiently. In our project, we rely on PyCharm's debugger for debugging machine learning models and algorithms.

3. **Web Development Support**: PyCharm provides comprehensive support for web development, including HTML, CSS, and JavaScript support. It seamlessly integrates with popular web frameworks such as Django and Flask, enhancing the development experience for web developers. Although our project does not primarily focus on web development, PyCharm's web development support can be beneficial for building user interfaces or web-based applications related to our project.

CHAPTER-9

SYSTEM TESTING

9.1 Literature on Testing

System testing for crop yield prediction uses some methods. following are some of the methods and approaches used in estimating yield:

- Environmental conditions and crop-growth models.
- Satellite monitoring and meteorological statistical models.
- Remote sensing technologies.
- Artificial Intelligence and Machine Learning models.

9.1.1 Unit Testing

Unit testing is a software development process that involves testing the smallest testable parts of an application, called units. The main objective of unit testing is to isolate written code to test and determine if it works as intended. Unit testing is an important step in the development process, and if done correctly, it can detect early flaws in code which may be more difficult to find in later testing stages.

Unit testing is typically developer-centric and aims to identify defects in code components before they impact the broader system. Unit testing is a kind of white box testing, whereas Integration Testing is a kind of black-box testing. For Unit Testing, accessibility of code is required, as it tests the written code, while for Integration Testing, access to code is not required, since it tests the interactions and interfaces between modules.

9.2.1 Integration Testing

Integration testing is a crucial phase in the software development lifecycle where individual software modules are combined and tested as a group to ensure seamless operation within a subsystem or the entire system. This process is typically carried out by members of the software development team, including testers or developers, who are responsible for verifying that the integrated modules function correctly together. Integration testing is performed after the completion of unit testing, which focuses on testing individual modules in isolation.

Techniques:

1. **Top-down approach:** In this approach, testing begins at the highest-level module, and lower-level modules are replaced with stubs, which are simulated modules. This allows for testing the integration of modules from the top down, ensuring that higher-level functionalities are properly integrated and functioning.

2. **Bottom-up approach:** Conversely, the bottom-up approach starts with testing low-level modules, and higher-level modules are simulated using drivers. This approach verifies the integration of modules from the bottom up, ensuring that foundational functionalities are correctly integrated and operational.

3. **Big bang approach:** This approach involves integrating all modules at once, which can be complex to manage but offers the advantage of testing the system as a whole. It requires comprehensive planning and coordination to ensure all modules are integrated correctly and functioning as expected.

9.3.1 Acceptance Testing

Acceptance testing focuses on verifying whether the entire software system meets the functional and non-functional requirements specified by the client or end-users. This phase involves various stakeholders, including clients, end-users, or independent testers, who assess the system's compliance with predefined criteria. Acceptance testing is performed after integration and system testing have been completed.

Types:

1. **User Acceptance Testing (UAT):** UAT simulates real users interacting with the system to evaluate its usability and determine if it meets their needs. This type of testing helps ensure that the software aligns with user expectations and requirements.

2. **Business Acceptance Testing (BAT):** BAT focuses on validating whether the system satisfies the business goals and requirements defined by the client. It assesses the software's functionality from a business perspective to ensure it meets organizational objectives.

3. **Contract Acceptance Testing**: Contract acceptance testing verifies that the system fulfills the functionalities and features outlined in the contract between the developer and the client. This type of testing ensures contractual obligations are met and provides a basis for resolving any discrepancies between the agreed-upon requirements and the delivered software.

9.2 Test cases on Project

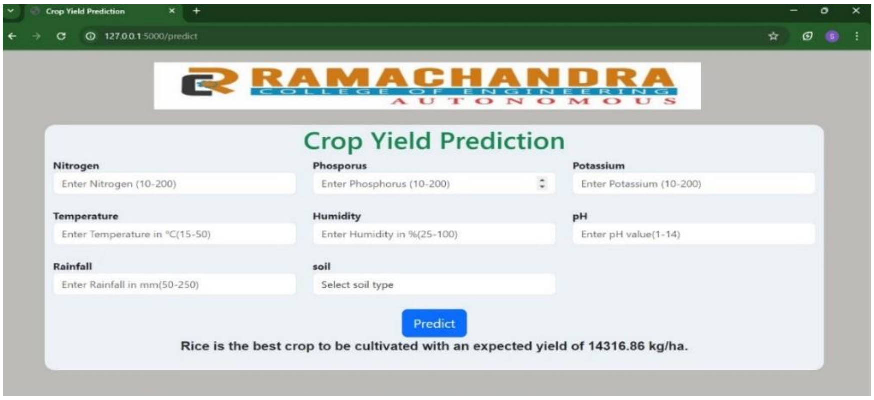
Test Case – 1

N=81,P=41,K=38, Temperature=22.67, Humidity=83.72, Ph=7.52, Rainfall=200.91, Soil=Black

Output:

Suggest crop is: Rice

Predict Yield Is: 14316.86 Kg/Ha.



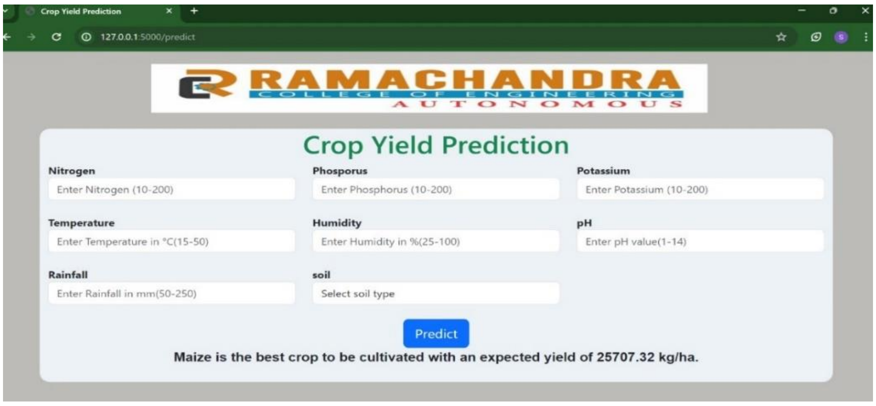
Test Case – 2

N=60, P=54, K=19, Temperature=18.74, Humidity= 62.49, Ph=6.41, Rainfall=70.23, Soil=Red

Output:

Suggest crop is: Maize

Predict yield Is: 25707.32 Kg/Ha.



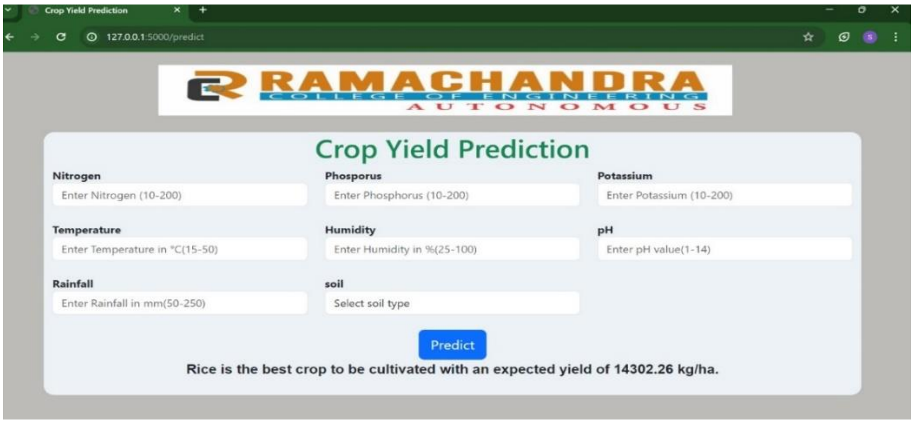
Test Case – 3

N=98, P=47, K=37, Temperature=23.48, Humidity= 81.33, Ph=7.37, Rainfall=224.05, Soil=Black

Output:

Suggest crop is: Rice

Predict yield Is: 14302.26 Kg/Ha.



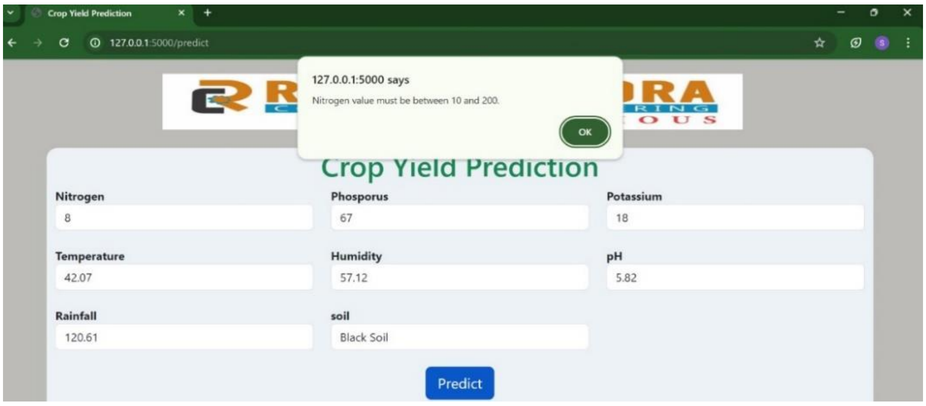
Test Case – 4

N=8, P=67, K=18, Temperature=42.07, Humidity= 57.12, Ph=5.82, Rainfall=120.61, Soil=Black

Output:

(Input Error)

Nitrogen value must be between 10 and 200.



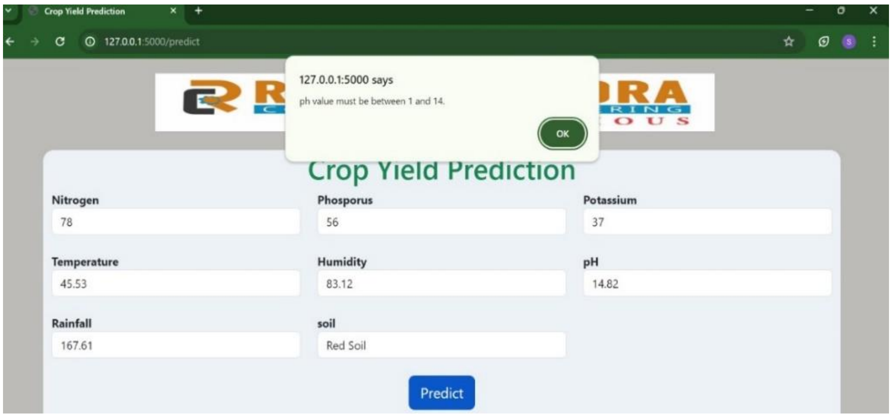
Test Case – 5

N=78, P=56, K=37, Temperature=45.53, Humidity= 83.12, Ph=14.82, Rainfall=167.61, Soil=Red

Output:

(Input Error)

pH value must be between 1 and 14.



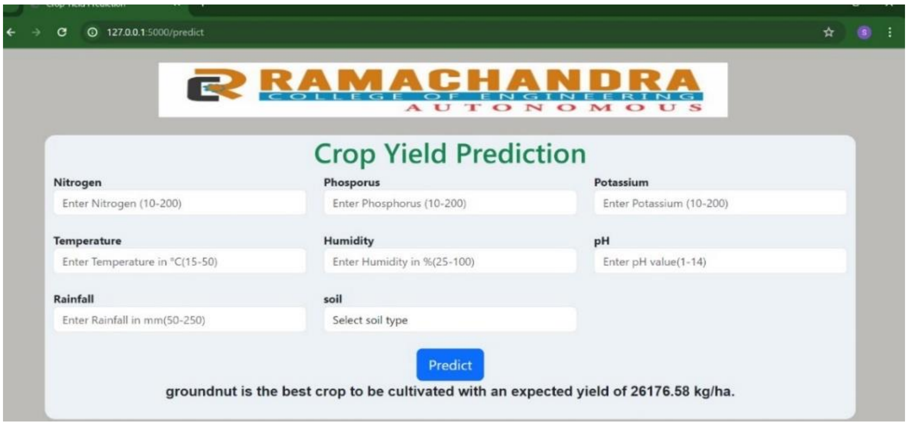
Test Case – 6

N=90, P=50, K=35, Temperature=26.48, Humidity= 71.33, Ph=6.37, Rainfall=244.15, Soil=Red

Output:

Suggest crop is: Ground Nut

Predict yield is: 26176.58 kg/ha.



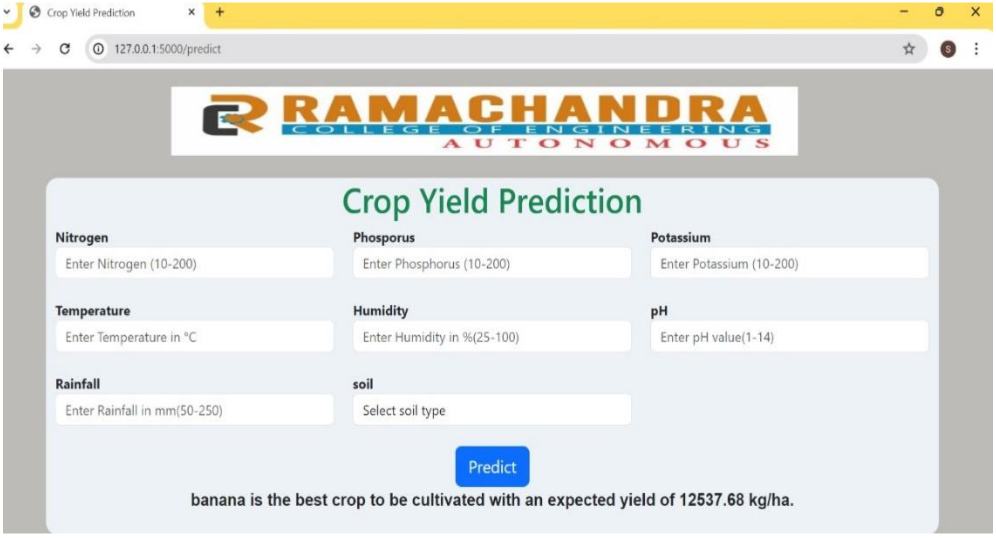
Test Case – 7

N=200, P=140, K=140, Temperature=35, Humidity= 56, Ph=6, Rainfall=89, Soil=Black

Output:

Suggest crop is: Banana

Predict yield is: 15615.33 kg/ha.



CHAPTER-10

CODING

Coding in Google Colab:

```
import numpy as np
import pandas as pd
crop = pd.read_csv("cs-1.csv")
crop.head()

#Display the first few rows using the head() function
crop.dtypes
crop.shape

#Number of rows and columns
crop.info()

#info()-information about the data types, non-null counts, and memory usage.
crop.isnull().sum()
crop.duplicated().sum()

#This will display the number of duplicate rows in the DataFrame crop
crop.describe()

#describe() method in pandas generates descriptive statistics of the numerical columns
in a DataFrame.
corr=crop.corr()
corr

#returns a correlation matrix.
crop_dict = {
   'rice': 1,
   'maize': 2
}
crop['crop_num']=crop['label'].map(crop_dict)
```

```
#crop['crop_num']-It will create a new column called crop num.
#map()-function is used to apply this mapping to each value in the 'label' column.

crop['crop_num'].value_counts()
X = crop.drop(['crop_num','label','yield'],axis=1)
y = crop[['crop_num','yield']]

#features-X
#target variable -y

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler
ohe = OneHotEncoder(drop='first')
scale = StandardScaler()
preprocesser = ColumnTransformer(
    transformers = [
        ('StandardScale', scale, [0, 1, 2, 3,4,5,6]),
        ('OHE', ohe, [7]),
    ],
    remainder='passthrough'
)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
X_train_dummy = preprocesser.fit_transform(X_train)
X_test_dummy = preprocesser.transform(X_test)
from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()

X_train_dummy = ms.fit_transform(X_train_dummy)
X_test_dummy = ms.transform(X_test_dummy)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
sc.fit(X_train_dummy)
X_train_dummy = sc.transform(X_train_dummy)
X_test_dummy = sc.transform(X_test_dummy)
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import ExtraTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score

# create instances of all models
models = {
    'Logistic Regression': LogisticRegression(),
    'Naive Bayes': GaussianNB(),
    'Support Vector Machine': SVC(),
    'K-Nearest Neighbors': KNeighborsClassifier(),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'Bagging': BaggingClassifier(),
    'AdaBoost': AdaBoostClassifier(),
    'Gradient Boosting': GradientBoostingClassifier(),
    'Extra Trees': ExtraTreeClassifier(),
}
y_train_crop_num = y_train['crop_num']
y_train_yield = y_train['yield']
for name, model in models.items():
    model.fit(X_train_dummy, y_train_crop_num)
    y_pred = model.predict(X_test_dummy)
    accuracy = accuracy_score(y_test['crop_num'], y_pred)
```

```
    print(f"{name} for crop number prediction with accuracy: {accuracy}")
for name, model in models.items():

# Train the model
    model.fit(X_train_dummy, y_train_yield)

 # Make predictions
    y_pred = model.predict(X_test_dummy)

 # Calculate accuracy
    accuracy = accuracy_score(y_test['yield'], y_pred)
    # Print the accuracy
    print(f"{name} for yield prediction with accuracy: {accuracy}")
from sklearn.ensemble import RandomForestClassifier

# Separate y_train into crop number and yield
y_train_crop_num = y_train['crop_num']
y_train_yield = y_train['yield']
# Train RandomForestClassifier for crop number prediction
rfc_crop_num = RandomForestClassifier()
rfc_crop_num.fit(X_train_dummy, y_train_crop_num)
# Make predictions for crop number
ypred_crop_num = rfc_crop_num.predict(X_test_dummy)

# Calculate accuracy for crop number prediction
accuracy_crop_num = accuracy_score(y_test['crop_num'], ypred_crop_num)
print(f"Random Forest Classifier for crop number prediction with accuracy:
{accuracy_crop_num}")

# Train RandomForestClassifier for yield prediction
rfc_yield = RandomForestClassifier()
rfc_yield.fit(X_train_dummy, y_train_yield)

# Make predictions for yield
ypred_yield = rfc_yield.predict(X_test_dummy)
```

# Calculate accuracy for yield prediction
accuracy_yield = accuracy_score(y_test['yield'], ypred_yield)
print(f"Random Forest Classifier for yield prediction with accuracy: {accuracy_yield}")
from sklearn.ensemble import RandomForestRegressor
regression_model = RandomForestRegressor()
regression_model.fit(X_train_dummy, y_train_yield)
from sklearn.ensemble import RandomForestRegressor

```
# Define the recommendation function
def recommendation(N, P, k, temperature, humidity, ph, rainfall, soil):

    # Prepare features
    features = np.array([[N, P, k, temperature, humidity, ph, rainfall, soil]])

    # Transform features
    transformed_features = preprocesser.transform(features)
    transformed_features = ms.transform(transformed_features)
    transformed_features = sc.transform(transformed_features)

    # Predict crop label
    predict_crop = rfc_crop_num.predict(transformed_features)[0]

    # Predict crop yield using the regression model
    predict_yield = regression_model.predict(transformed_features)[0]
    return predict_crop, predict_yield
```

 # Example input values

N = 71

P = 54

k = 16

Temperature = 22.6136

Humidity = 63.69071

ph = 5.749914

Rainfall = 87.759554

soil = 'red'

```
# Get recommendation
predict_crop, predict_yield = recommendation(N, P, k, temperature, humidity, ph, rainfall, soil)

# Output recommendation
crop_dict = {1: "Rice", 2: "Maize"}

# Check if predict_crop is a single value before accessing crop_dict
if isinstance(predict_crop, np.ndarray):
    predict_crop = predict_crop[0]

if predict_crop in crop_dict:
    crop = crop_dict[predict_crop]
    print("{} is the best crop to be cultivated with an expected yield of {}.".format(crop, predict_yield))
else:
    print("Sorry, we are not able to recommend a proper crop for this environment.")
import pickle
pickle.dump(rfc,open('model.pkl','wb'))
pickle.dump(preprocesser,open('preprocessor.pkl','wb'))
pickle.dump(ms,open('minmaxscaler.pkl','wb'))
pickle.dump(sc,open('standscaler.pkl','wb'))
```

Coding in PyCharm

```
from flask import Flask,request,render_template
import numpy as np
import pandas
import sklearn
import pickle

# importing model
model = pickle.load(open('model.pkl','rb'))
preprocessor = pickle.load(open('preprocessor.pkl','rb'))
sc = pickle.load(open('standscaler.pkl','rb'))
ms = pickle.load(open('minmaxscaler.pkl','rb'))

# creating flask app
app = Flask(__name__)

@app.route('/')
def index():
    return render_template("index.html")
@app.route("/predict",methods=['POST'])
def predict():
    N = request.form['Nitrogen']
    P = request.form['Phosporus']
    K = request.form['Potassium']
    temp = request.form['Temperature']
    humidity = request.form['Humidity']
    ph = request.form['Ph']
    rainfall = request.form['Rainfall']
    soil=request.form['Soil']
    feature_list = np.array([[N, P, K, temp, humidity, ph,
rainfall,soil]],dtype="object")
    transformed_features = preprocessor.transform(feature_list)
    transformed_features = ms.transform(transformed_features)
    transformed_features = sc.transform(transformed_features)
    result = model.predict(transformed_features).reshape(1, -1)
    crop_dict = {1: "Rice", 2: "Maize"}
    if isinstance(result, np.ndarray):
        predict_crop = result[0][0]
        predict_yield = result[0][1]
    if predict_crop in crop_dict:
        crop = crop_dict[predict_crop]
        result = "{} is the best crop to be cultivated with an expected yield of {}
kg/ha.".format(crop, predict_yield)
    else:
        result = "Sorry, we are not able to recommend a proper crop for this
environment."
    return render_template('index.html',result= result)

# python main
if __name__ == "__main__":
    app.run(debug=True)
```

Coding in webpage:

```
<!doctype html>
<html lang="en">
  <head>
    <title>Crop Yield Prediction</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN
9+nJOZ" crossorigin="anonymous">
  </head>
  <style>
      h1 {
         color: mediumseagreen;
         text-align: center;
      }
      h2 {
         color: white;
         text-align: left;
      }
      .warning {
         color: red;
         font-weight: bold;
         text-align: center;
      }
      .card{
      margin-left:410px;
      margin-top: 200px;
      color: white;
      }
      .container{
      background:#edf2f7;
      font-weight: bold;
      padding-bottom:10px;
      border-radius: 15px;
```

```
        }
       .container1 {
         padding-bottom:10px;
         padding-top:20px;
         padding-left:230px;
         font-size: 24px;
         color: white;
         }
       #blink{
       font-size: 20px;
       font-weight: bold;
       font-family: sans-serif;
       text-align:center;
       }
     </style>

   <body style="background:#BCBBB8">

   <div class="container1">
     <img src="https://media.licdn.com/dms/image/D5616AQEHQErXk-
g6oQ/profile-displaybackgroundimage-
shrink_200_800/0/1690984507532?e=2147483647&v=beta&t=RTvXSwAH7yKmh
6vB_kZaYGVF6lRUgr7zMZsRalIclOQ" alt="logo" width="800" height="80">
   </div>
   <div class="container my-3 mt-3">
       <h1 class="text-success">Crop Yield Prediction <span class="text-
success"></span></h1>
<!--      adding form-->
       <form action="/predict" method="POST" onsubmit="return validateForm()">
        <div class="row">
          <div class="col-md-4">
            <label for="Nitrogen">Nitrogen</label>
            <input type="number" id="Nitrogen" name="Nitrogen" placeholder="Enter
Nitrogen (10-200)" class="form-control" required step="1" onkeydown="javascript:
return ['Backspace','Delete','Tab','ArrowLeft','ArrowRight','tab'].includes(event.code)
```

```
? true : !isNaN(Number(event.key)) && event.code!=='Space'">
          </div>
          <div class="col-md-4">
            <label for="Phosporus">Phosporus</label>
            <input type="number" id="Phosporus" name="Phosporus"
placeholder="Enter Phosphorus (10-200)" class="form-control" required step="1"
onkeydown="javascript: return
['Backspace','Delete','Tab','ArrowLeft','ArrowRight'].includes(event.code) ? true :
!isNaN(Number(event.key)) && event.code!=='Space'">
          </div>
          <div class="col-md-4">
            <label for="Potassium">Potassium</label>
            <input type="number" id="Potassium" name="Potassium"
placeholder="Enter Potassium (10-200)" class="form-control" required step="1"
onkeydown="javascript: return
['Backspace','Delete','Tab','ArrowLeft','ArrowRight'].includes(event.code) ? true :
!isNaN(Number(event.key)) && event.code!=='Space'">
          </div>
        </div>
         <div class="row mt-4">
            <div class="col-md-4">
              <label for="Temperature">Temperature</label>
              <input type="number" id="Temperature" name="Temperature"
placeholder="Enter Temperature in °C(15-50)" class="form-control" step="0.01">
            </div>
            <div class="col-md-4">
              <label for="Humidity">Humidity</label>
              <input type="number" id="Humidity" name="Humidity"
placeholder="Enter Humidity in %(25-100)" class="form-control" step="0.01">
            </div>
            <div class="col-md-4">
              <label for="pH">pH</label>
              <input type="number" id="Ph" name="Ph" placeholder="Enter pH
value(1-14)" class="form-control" step="0.01">
            </div>
```

```
        </div>

        <div class="row mt-4">
          <div class="col-md-4">
            <label for="Rainfall">Rainfall</label>
            <input type="number" id="Rainfall" name="Rainfall" placeholder="Enter
Rainfall in mm(50-250)" class="form-control" required step="0.01">
          </div>
          <div class="col-md-4">
            <label for="Soil">soil</label>
            <select id="Soil" name="Soil"  class="form-control">
              <option value="">Select soil type</option>
                <option value="black">Black Soil</option>
                <option value="red">Red Soil</option>
             </select>
          </div>
        </div>

      <!-- Rest of the form -->
      <div class="row mt-4">
        <div class="col-md-12 text-center">
          <button type="submit" class="btn btn-primary btn-lg">Predict</button>
        </div>
      </div>
    </form>

       <div class="card-body">
        <p id="blink" style="text-align:center">{{ result }}</p>
        <script type="text/javascript">
          var blink = document.getElementById('blink');
          setInterval(function () {
            blink.style.opacity = (blink.style.opacity == 0 ? 1 : 0);
          }, 1000);
        function validateForm() {
            var nitrogen = document.getElementById("Nitrogen").value;
```

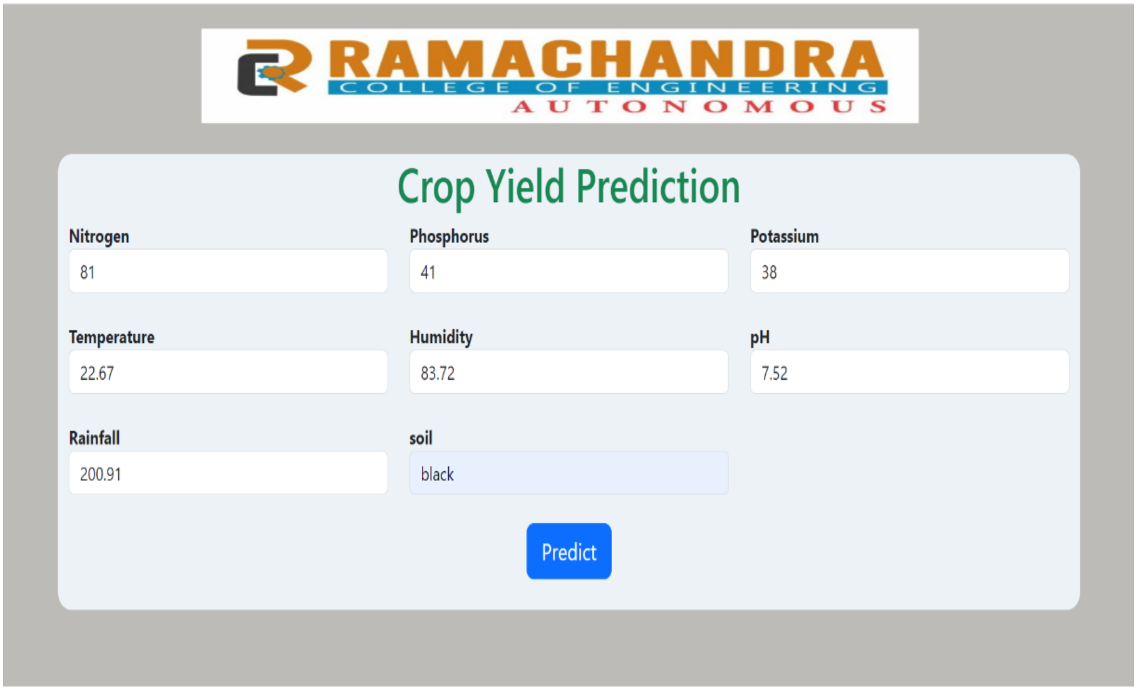
```
var phosporus = document.getElementById("Phosporus").value;
var potassium = document.getElementById("Potassium").value;
var temperature = document.getElementById("Temperature").value;
var humidity = document.getElementById("Humidity").value;
var ph = document.getElementById("Ph").value;
var rainfall = document.getElementById("Rainfall").value;


if (nitrogen < 10 || nitrogen > 200) {
  alert("Nitrogen value must be between 10 and 200.");
  return false;
}

if (phosporus < 10 || phosporus > 200) {
  alert("Phosphorus value must be between 10 and 200.");
  return false;
}

if (potassium < 10 || potassium > 200) {
  alert("Potassium value must be between 10 and 200.");
  return false;
}
if (temperature < 10 || temperature > 50) {
  alert("Temperature value must be between 10 and 50.");
  return false;
}
if (humidity < 25 || humidity > 100) {
  alert("Humidity value must be between 25 and 100.");
  return false;
}
if (ph < 1 || ph > 14) {
  alert("ph value must be between 1 and 14.");
  return false;
}
if (rainfall < 50 || rainfall > 250) {
```

```
            alert("rainfall value must be between 50 and 250.");
            return false;
          }

          // If all validations pass, return true to submit the form
          return true;
        }
      </script>
    </div>
  </div>

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQ
SZe" crossorigin="anonymous"></script>
  </body>
</html>
```

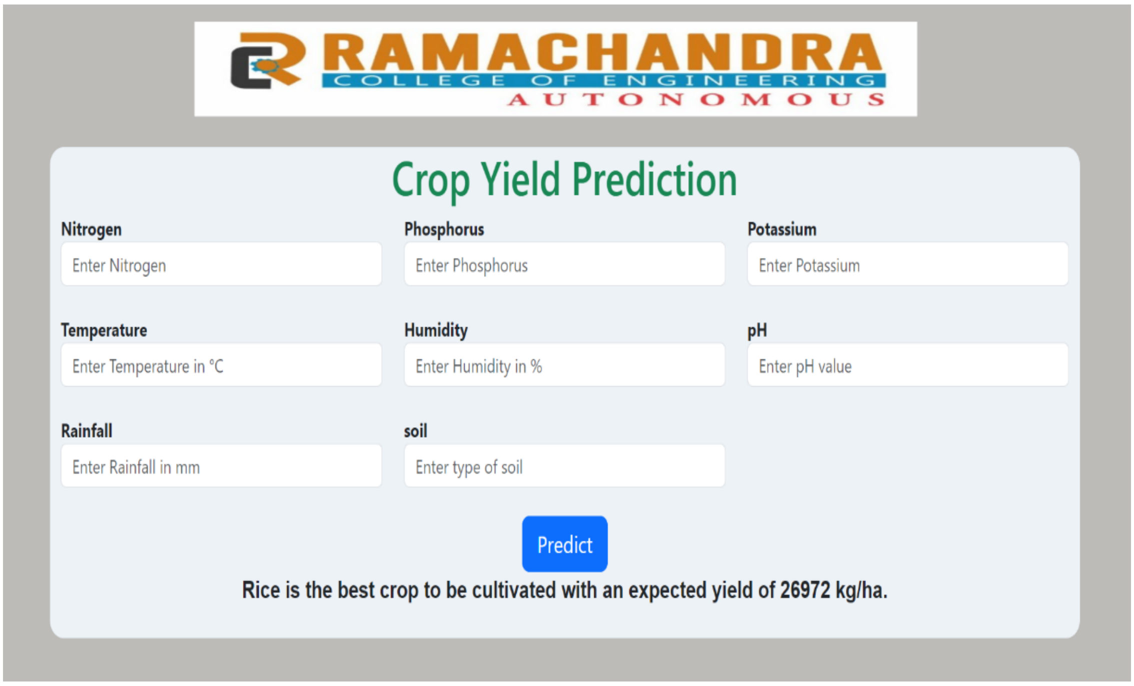
CHAPTER-11

RESULTS

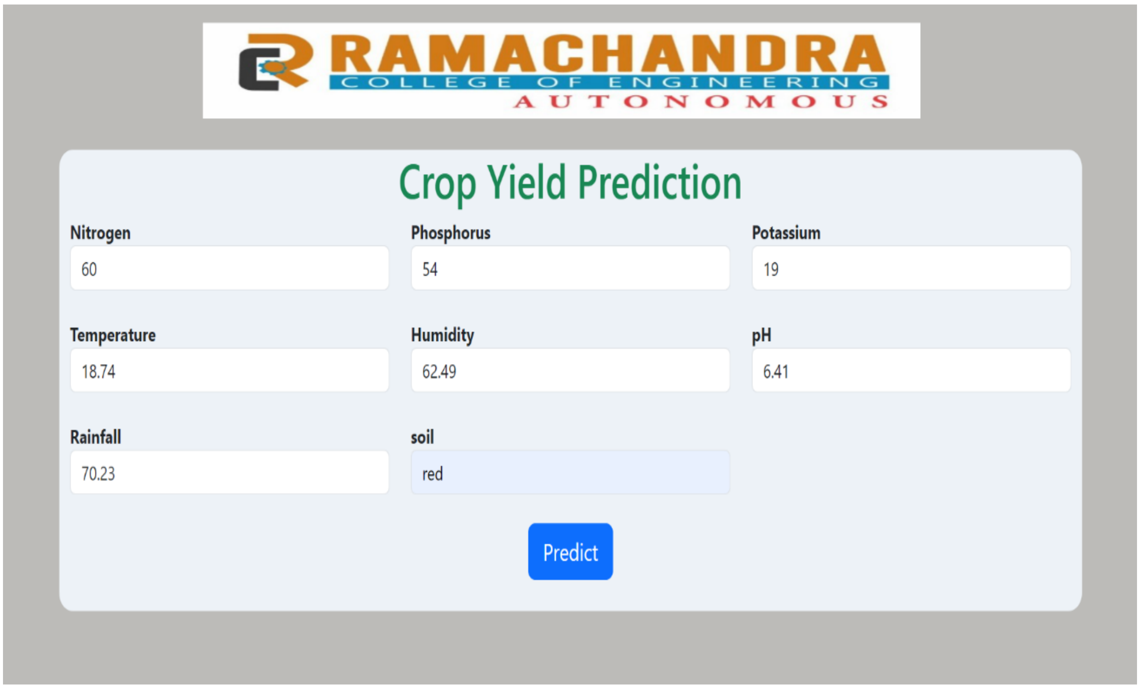
11.1 Screenshots



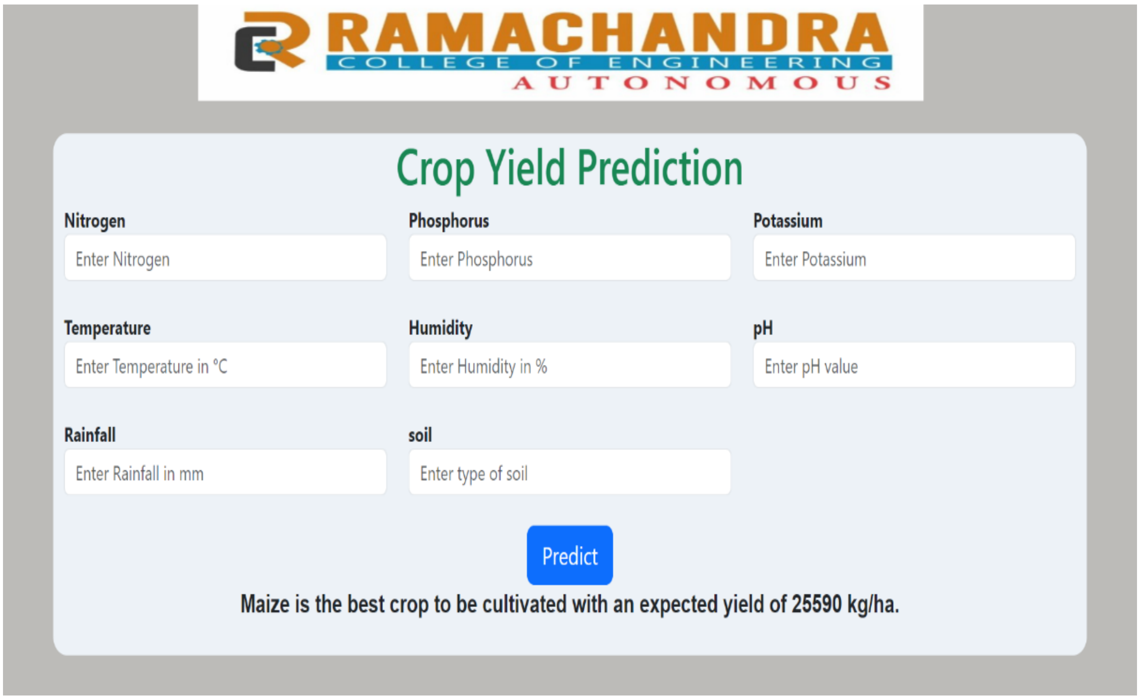
Giving inputs for the prediction



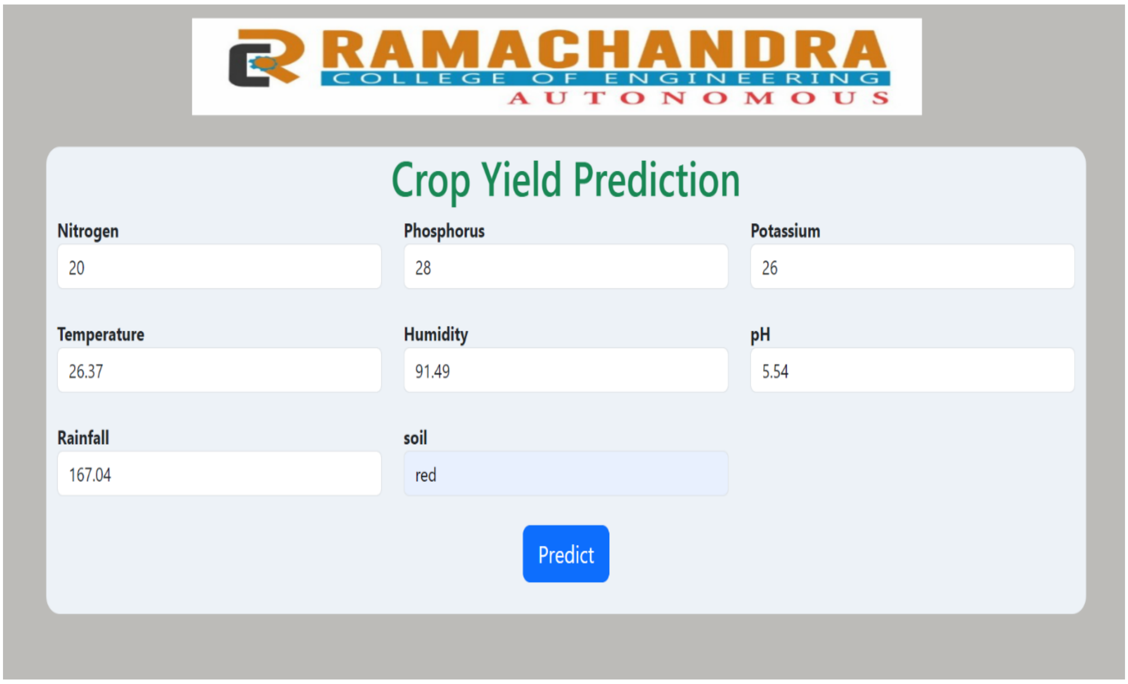
Output for the given inputs



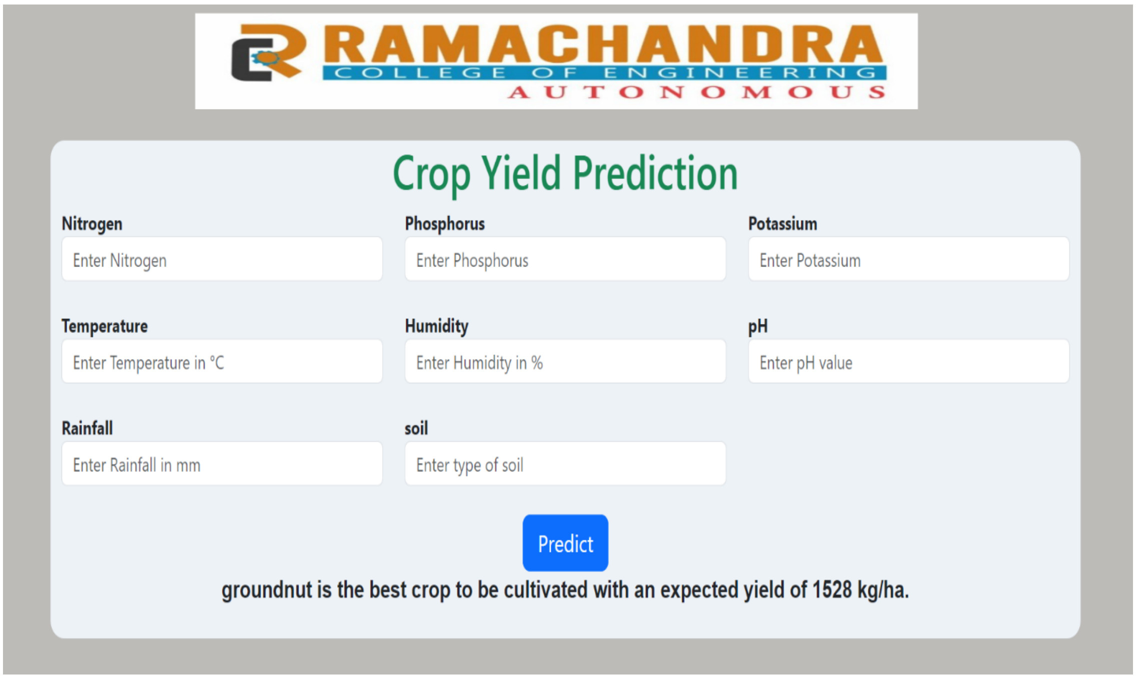
Given inputs for the another prediction



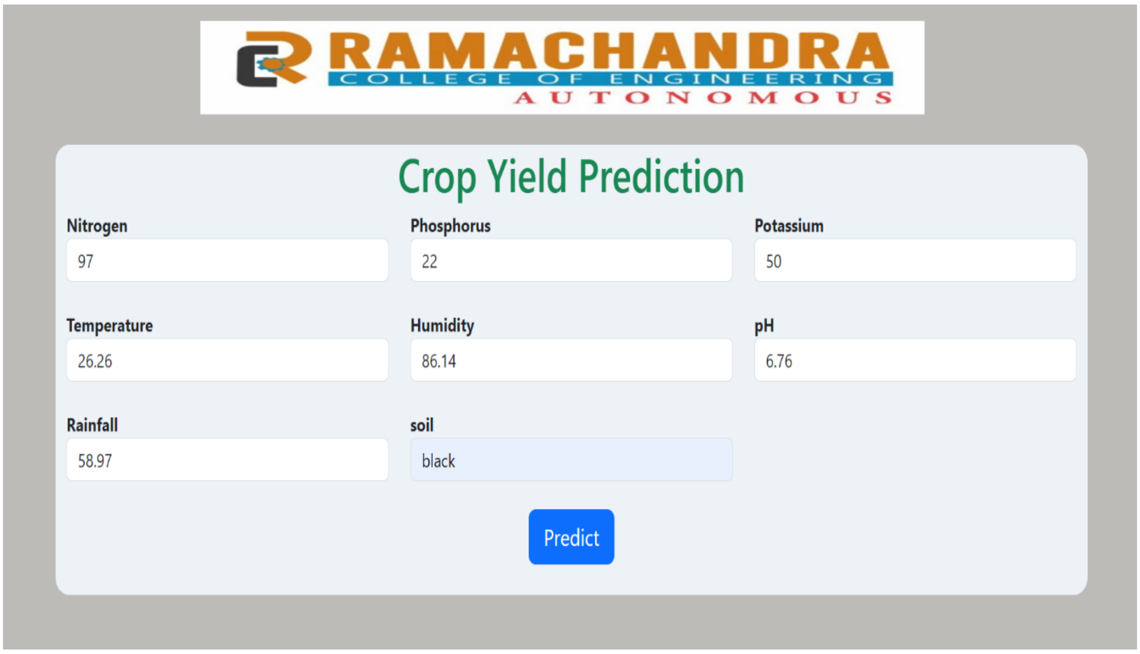
Output for the given inputs



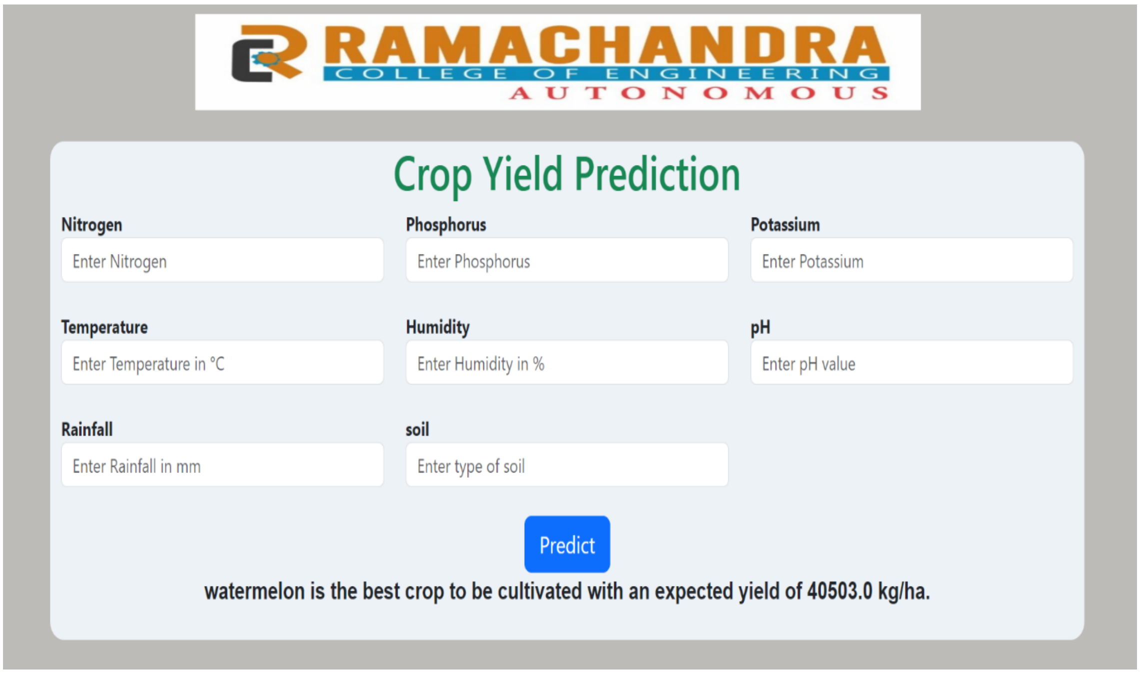
Given inputs for the another prediction



Output for the given inputs



Given inputs for the another prediction



Output for the given inputs

11.2 Conclusion & Limitations

Agriculture is the field which helps in economic growth of our country. But this is lacking behind in using new technologies of machine learning. Hence our farmers should know all the new technologies of machine learning and other new techniques. These techniques help in getting maximum yield of crops.

Many techniques of machine learning are applied on agriculture to improve yield rate of crops. These techniques also help in solving problems of agriculture. We can also get the accuracy of yield by checking for different methods. Hence we can improve the performance by checking the accuracy between different crops.

Crop yield prediction is a complex process which relies on several different factors including weather, soil, etc. In this paper, we predict the crop yield using weather and soil parameters. The system incorporates regression techniques to estimate yield and multi-class classification to predict type of the crop. Among the used models for yield prediction, Random Forest Regression gives best results.

1. **Data Availability and Quality:** Limited availability of high-quality data, especially in regions with less developed agricultural monitoring systems, can hinder the accuracy of crop yield predictions. Historical data may be incomplete, inconsistent, or biased, leading to challenges in training robust ML models.
2. **Data Variability and Complexity:** Agricultural systems are influenced by various factors such as weather conditions, soil properties, pest infestations, and farming practices. Capturing and incorporating all relevant variables into ML models is challenging. The spatial and temporal variability of agricultural data presents difficulties in generalizing predictions across different regions and time periods.
3. **Model Overfitting and Generalization:** ML models trained on historical data may overfit to specific patterns present in the training dataset, resulting in poor generalization to unseen data. Ensuring that ML models generalize well across different years, regions, and agricultural practices requires careful model selection, feature engineering, and validation techniques.
4. **Limited Scalability:** Scalability of ML models to large geographical areas or entire crop production systems can be challenging due to computational constraints and data processing requirements. Deploying and maintaining ML models at scale may require

significant infrastructure and resources, particularly in resource-constrained agricultural settings.

5. **Integration with Existing Farm Management Practices:** Implementing machine learning models into existing farm management practices requires seamless integration with existing tools and workflows. Disruptive changes or complex interfaces can discourage adoption by farmers.
6. **Addressing Social and Ethical Concerns:** The use of machine learning in agriculture raises social and ethical concerns. Bias in training data can lead to discriminatory outcomes for certain farmers or regions. Data privacy and security are also important considerations, as agricultural data can be sensitive.
7. **Infrastructure and Connectivity:** Deploying machine learning models in remote areas can be hindered by a lack of reliable internet connectivity and access to computing power. Developing lightweight models and exploring offline inference techniques can help address this challenge.
8. **Cost and Return on Investment:** The cost of implementing machine learning solutions in agriculture, including data collection, model development, and deployment, needs to be justified by the potential return on investment for farmers. Demonstrating the clear economic benefits is crucial for widespread adoption.

11.3 Future Scope

Predicting crop yield accurately is crucial for ensuring food security, optimizing agricultural practices, and mitigating the effects of climate change. Here are some potential future enhancements for crop yield prediction:

- **Machine Learning Integration:** Implement advanced algorithms like deep learning and neural networks to enhance crop yield prediction accuracy by analyzing large datasets and recognizing complex patterns.

- **Satellite Imagery Integration:** Utilize high-resolution satellite images to monitor crop growth and detect stressors, aiding in yield prediction through image processing techniques.

- **Multi-Source Data Fusion:** Combine diverse data sources such as weather, soil properties, and crop history to construct robust crop yield prediction models, managing data diversity effectively.

- **IoT and Sensors:** Employ IoT devices and sensors in fields to gather real-time environmental and crop health data, integrating this with predictive analytics for accurate forecasts.

- **Climate Scenario Analysis:** Incorporate climate models to evaluate climate change impacts on crop yields, offering insights for adaptive strategies.

- **Spatial Models:** Develop models considering local soil, topography, and microclimates for site-specific yield predictions and precision agriculture.

CHAPTER-12

REFERENCES & BIBLIOGRAPHY

1. Alpaydın, “Introduction to machine learning, second edition.” MIT Press, 2010. ISBN: 978-0-262-01243-0.

2. Sabitha, “A study on sectorial contribution of GDP in India from 2010 to 2019”, AJEBA, 19, no. 1, pp. 18-31, January 2020. Article no. AJEBA. 62227.

3. Jain A., “Analysis of growth and instability in the area, production, yield, and price of rice in India”, Journal of Social Change and Development, vol. 2, pp.

4. Sangeeta, Shruthi G. “Design and implementation of crop yield prediction model in ” International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 4, pp. 305-310, Apr. 2020.

5. Johnson LK, Bloom JD, Dunning RD, Gunter CC, Boyette MD, Creamer NG, “Farmer harvest decisions and vegetable loss in primary production. Agricultural Systems”, 176, pp. 102672, November 2019.

6. Sharma A, Jain A, Gupta P, Chowdary V. Machine learning applications for precision agriculture: A comprehensive review. IEEE Access. 2020 Dec 31;9:4843-73.

7. Meshram V, Patil K, Meshram V, Hanchate D, Ramkteke SD. Machine learning in agriculture domain: A state-of-art survey. Artificial Intelligence in the Life Sciences. 2021 Dec 1;1:100010.

8. Reddy, D. J., & Kumar, M. R. (2021). Crop Yield Prediction using Machine Learning Algorithm. 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). doi:10.1109/iciccs51141.2021.9432236.

9. Ranjini B Guruprasad, Kumar Saurav, Sukanya Randhawa, “Machine learning methodologies for paddy yield Estimation in India: a case study”.

10. S. P. Raja, B. Sawicka, Z. Stamenkovic and G. Mariammal, “Crop prediction based on characteristics of the agricultural environment using various feature selection techniques and classifiers,” IEEE Access, vol. 10, pp. 23625-23641, 2022, doi: 10.1109/ACCESS.2022.3154350.