VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka: 590018

Project Report On

CROPS PREDICTION USNIG MACHINE LEARNING TECHNIQUES

Submitted by

BHUVAN
VENKAT PAWAN G

Submitted in partial fulfilment for the award of Degree of

BACHELOR OF ENGINEERING
In

COMPUTER SCIENCE ENGINEERING

Under the Guidance of

Prof. Manju D
Assistant Professor
Department of ECE
SVCE, Bengaluru

Dr. Jijesh J J
HOD
Department of ECE
SVCE, Bengaluru

Dr. Latha M S
HOD
Department of CIVIL
SVCE, Bengaluru

Prof. Archana M
Assistant Professor
Department of CSE
SVCE, Bengaluru

DEPARTMENT OF CSE

SRI VENKATESHWARA COLLEGE OF ENGINEERING

(Affiliated to VTU, Belagavi, NBA* Accredited, Approved by AICTE New Delhi)Kempegowda International Airport Road, Vidyanagar, Bengaluru – 562157.

2022 – 2023

SRI VENKATESHWARA COLLEGE OF ENGINEERING

Vidyanagar, Bengaluru - 562157

Department of CSE

CERTIFICATE

This is to certify that the Project Work entitled

"CROPS PREDICTION USING MACHINE LEARNING TECHNIQUES"

Carried out by

BHUVAN
VENKAT PAWAN G

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi, during the year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The Project Report has been approved as it satisfies the academic requirements in respect of the Project Work prescribed for the said Degree.

Signature of the Guide	**Signature of the Guide**	**Signature of the Guide**	**Signature of the Guide**
(Mr. Manju D)	(Dr. Jijesh J J)	(Mrs. Archana M)	(Dr. Latha M S)
Asst. Professor	HOD	Asst. Professor	HOD
ECE, SVCE	ECE, SVCE	CSE, SVCE	CIVIL, SVCE

Signature of the HOD	**Signature of the Principal**
(Dr. Hema M S)	(Dr. Nageswara Guptha M)
CSE, SVCE	SVCE

External Viva -Voce

Name of the Examiners	**Signature with date**
1.	
2.	

DECLARATION

We **BHUVAN, and VENKAT PAWAN G** student of final semester B.E in Computer Science Engineering, Sri Venkateshwara College of Engineering, Bengaluru, hereby declare that the dissertation work entitled **“CROPS PREDICTION USING MACHINE LEARNING TECHNIQUES”** has been carried out under supervision **Mr. Manju D,** Assistant professor and **Dr. Jijesh JJ,** HOD, Department of Electronics & Communication Engineering and **Mrs. Archana M,** Assistant professor, Department of Computer Science Engineering and **Dr. Latha M S,** HOD, Department of Civil, SVCE, Bengaluru, the partial fulfilment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science Engineering by Visvesvaraya Technological University, Belagavi during the academic year 2022-23. Further, the matter embodied in the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

Date: Place: Bengaluru

BHUVAN
VENKAT PAWAN G

ACKNOWLEDGEMENT

The euphoria and complacency of completing this technologically advanced project will not be complete until I/we think all the people who have helped us in complete this enthusiastic work. Submission of this project report marks a milestone in our academic career.
After years of schooling, there are lots of people who have helped us, either directly or indirectly,in our achievements in life. It is a pleasure to acknowledge such people.

We are grateful to our Chairman, **Sri. V Muniyappa**, Chief Executive Director, **Dr. Shashidhar Muniyappa**, and Management for providing us with the necessary equipment & assistance to complete this project.

Our deepest thankfulness goes to our Principal, **Dr. Nageswara Gupta** for his continuous encouragement and support.

Our deepest thankfulness goes to our Dean Academics, **Dr. Poornima G R**, Department of Electronics and Communication Engineering. We do consider ourselves privileged to have had the opportunity to work under her guidance.

We would like to thank beloved **Dr. Hema M S**, Head of the Department, Department of ComputerScience Engineering for her encouragement and support.

Our deepest thankfulness goes to our project coordinator, **Mrs. Varsha Kulkarni**, Assistant Professor, Department of Computer Science Engineering. We do consider ourselves privileged tohave had the opportunity to work under her guidance.

We wish to express our sincere appreciation for our committee members who took the time to gothrough our project work. **Mr. Manju D,** Assistant professor and **Dr. jijesh JJ,** HOD, Department of Electronics & Communication Engineering and **Mrs. Archana M,** Assistant professor, Department of Computer Science Engineering and **Dr. Latha M S,** HOD, Department of Civil, our guides, who throughout remained a supportive person.

We would like to express our gratitude to the Professors and Lab instructors the of Department of Computer Science Engineering of their encouragement. Finally, we thank our parents who were on our side whenever we needed them and the almighty whose blessings are always there.

BHUVAN
VENKAT PAWAN G

INSTITUTE VISION

To be a premier institute for addressing the challenges in global perspective.

INSTITUTE MISSION

M-1: Nurture students with professional and ethical outlook to identify needs, analyze, design and innovate sustainable solutions through lifelong learning in service of society as individualor a team.

M-2: Establish State of the Art Laboratories and Information Resource center for education and research.

M-3: Collaborate with Industry, Government Organization and Society to align the curriculum and outreach activities.

DEPARTMENT VISION

Global Excellence with relevance in Computer Science and Engineering Education, Research and Development.

DEPARTMENT MISSION

M1: Accomplish academic achievement in Computer Science and Engineering through student-centered creative teaching learning, qualified faculty members, assessment and effective usage of ICT.

M2: Establish a Center of Excellence in a various verticals of computer science and engineering to encourage collaborative research and Industry-institute interaction.

M3: Transform the engineering students to socially responsible, ethical, technically competent and Value added professional or entrepreneur through holistic education.

Program Educational Objectives

PEO-1: Knowledge Computer Science and Engineering Graduates will have professional technical career in inter disciplinary domains providing innovative and sustainable solutions using modern tools.

PEO-2: Skills Computer Science and Engineering Graduates will have effective communication, leadership, team building, problem solving, decision making and creative skills.

PEO-3: Attitude Computer Science and Engineering Graduates will practice ethical responsibilities towards their peers, employers and society.

ABSTRACT

The Indian economy is based mostly on agriculture. India's economy relies heavily on agriculture, which contributes a significant amount of the country's gross domestic product and serves as the backbone of the country in general. However, artificial climate changes are reducing food production and crop forecasting, which will harm farmers' economies by resulting in a poor yield and impairing their capacity to predict future harvests. For agricultural practices to be optimized, food security to be ensured, and yield to be increased, crop prediction accuracy is essential. Due to their ability to analyze vast volumes of data and identify significant trends, machine learning algorithms have lately emerged as useful tools for crop prediction. This study aims to assess the accuracy of numerous machine learning approaches for predicting agricultural production, including Random Forest, Decision Tree, Logistic Regression, Support Vector Machine (SVM), Naive Bayes, and XGBoost. We also look at how environmental elements like temperature, rainfall, pH, nitrogen, phosphorus, and potassium levels affect crop growth.

To enhance the prediction models, we added environmental factors such temperature, rainfall, pH, nitrogen, phosphorus, and potassium as input characteristics. The relevance of each component in affecting crop yield was calculated using feature significance techniques. By examining the data, we discovered how these factors affect crop growth and yield. The algorithm under study, XGBoost, had the greatest accuracy and outstanding predictability for crop production estimation. Temperature, rainfall, and nitrogen were shown to have the greatest effects on crop growth, highlighting the significance of these variables in agricultural planning and decision-making. This study highlights the need of taking environmental factors into consideration when calculating production and offers helpful insights into the usage of machine learning algorithms for crop prediction. The results can assist farmers, agricultural policymakers, and researchers in ensuring long-term agricultural output, optimizing resource allocation, and improving crop management techniques.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Agriculture is one of the most crucial sectors in most countries, providing food and other essential resources to sustain life. With the growing population, there is an ever-increasing demand for food production. Crop yield prediction is a vital factor in agricultural management and production, as it helps farmers and policymakers plan for the future and make informed decisions about crop management practices.

Crop production forecasting has always involved manual observation and study of previous data. However, this approach takes a lot of time, is prone to mistakes, and is not appropriate for large-scale farming. Machine learning (ML) approaches have become a potential option for predicting agricultural yields as a result of technological improvements. Crop prediction using machine learning algorithms has emerged as a key area of agricultural research due to the rising worldwide need for food. Huge data sets may be analyzed by machine learning algorithms, which can then use the correlations and patterns found in the data to produce precise predictions. By predicting crop yield and supplying the right fertilizers and environmental conditions for optimum development, this method may be used to agriculture.

In this article, we will examine the many machine learning techniques, including random forest, SVM, logistic regression, naive bayes, xg boosting, and decision trees, that may be used for crop prediction and recommendation. We will also discuss the important factors affecting crop output, such as nitrogen, phosphorus, potassium levels, temperature, rainfall and pH. Nitrogen, phosphorous, and potassium are vital plant nutrients that are typically present in fertilizers. They are crucial in the development of leaves, stems, roots, and flowers. Nitrogen is in charge of leaf growth and overall plant vigor, phosphorus is in charge of root growth and the formation of flowers and fruits, and potassium is essential for overall plant health and stress tolerance. The concentrations of these nutrients in the soil can have a major impact on crop yield. Rainfall and temperature are other significant elements that influence crop development. various plants require various temperatures for optimum growth, and excessive temperatures can stress plants and reduce agricultural output. Rainfall is necessary to supply water to plants, and drought or an abundance of rain can significantly reduce agricultural productivity. Last but not least, soil pH is also crucial for crop growth. Various crops require varying pH values for optimum growth, and an imbalance in pH levels can alter how well plants absorb nutrients, resulting in lower yields. To predict crop yield and recommend the best fertilizers and environmental conditions, we need to gather data on these key attributes. Once the data has been collected, machine learning algorithms can be used to analyze the data and provide accurate predictions and recommendations.

We must preprocess the data by cleaning, converting, and normalizing the data in order to employ these machine learning methods for crop prediction. This phase is necessary to guarantee that the machine learning algorithms can analyze the data precisely and generate predictions and suggestions that are correct. Following preprocessing, we may apply machine learning methods to train the model using the preprocessed data. The trained model may then be used to forecast crop production and suggest the appropriate fertilizers and environmental factors for healthy growth.

The quality and quantity of data utilized to train the model will determine how accurate the forecasts and advice are. To guarantee that the forecasts and suggestions are reliable, high-quality data must be gathered and used. Farmers may increase crop output and use less fertilizer and other resources by employing machine learning algorithms for crop prediction and recommendation, such as random forest, SVM, logistic regression, naive bayes, xg boosting, and decision trees. We can provide precise forecasts and suggestions that can assist farmers in making knowledgeable decisions about crop management by examining the major factors that affect crop production, such as nitrogen, phosphorus, and potassium levels, temperature, rainfall, and pH. To make sure that the machine learning algorithms can effectively analyze the data and produce accurate predictions and suggestions, it is crucial to acquire and use high-quality data as well as to preprocess the data.

The use of ML techniques in crop yield prediction has several advantages over traditional methods. Firstly, it is faster and more efficient as it can analyze large datasets in a short period. Secondly, it is more accurate as it can identify patterns and relationships that may not be visible to the human eye. Finally, it can provide real-time predictions, allowing farmers and policymakers to make informed decisions quickly. Despite the potential benefits of ML techniques in crop yield prediction, there are still some challenges that need to be addressed. One of the major challenges is the availability and quality of data. To train ML models, large amounts of high-quality data are needed, and collecting such data can be costly and time-consuming. Additionally, the interpretability of ML models can be a challenge, as it can be difficult to understand how the model arrives at its predictions.

In conclusion, crop production prediction is a crucial issue in agriculture, and ML approaches have shown promise as a replacement for current practices. To increase the precision and interpretability of the machine learning models used to estimate crop yields, however, there are still issues that must be resolved, and further research is required.

1.2 Objectives

- Develop machine learning algorithms to accurately predict crop yields based on weather, soil, and market data.
- To design and implement a model in order to achieve an efficiency.
- To create a model for crops prediction by enhancing the precision of agricultural yield predictions using ML approaches.
- To Improve crop selection and recommendation for farmers.
- Reduce crop losses due to unfavorable environmental factors and increase agricultural productivity and yield.

1.3 Advantages

- Improved accuracy: Machine learning models can analyze large volumes of data and identify patterns that humans may overlook, leading to more accurate crop predictions.
- Timely decision-making: Machine learning models can provide real-time or near real-time predictions, enabling farmers to make informed decisions promptly.
- Increased productivity: Accurate crop predictions can help farmers optimize their practices, leading to increased yields and higher productivity.
- Resource efficiency: By predicting crop yields and resource requirements, machine learning models can help farmers optimize the use of water, fertilizers, and pesticides, reducing waste and environmental impact.

- Risk mitigation: Crop prediction models can assist in identifying potential risks such as pests, diseases, or adverse weather conditions, allowing farmers to take preventive measures to minimize losses.
- Cost-effective: Accurate crop predictions can help farmers optimize their resources and reduce unnecessary expenses, resulting in cost savings.
- Sustainability: By optimizing resource utilization, reducing waste, and minimizing environmental impact, crop prediction using machine learning can contribute to sustainable farming practices.

CHAPTER 2

LITERATURE SURVEY

[1] "Crop yield prediction using machine learning: A review," Journal of Agricultural Science and Technology, vol. 23, no. 4, pp. 901-913, 2021.

Authors: N Devi and V.K.Gupta

Abstract: Predicting crop yields is one of the most important tasks in agriculture since it may help farmers make management decisions. Machine learning (ML) techniques have lately been used for predicting agricultural productivity due to their ability to analyze vast amounts of data and provide exact projections. In this review, we present a summary of the most recent research on predicting agricultural productivity using ML techniques. We discuss the many machine learning (ML) approaches—including decision trees, random forests, support vector machines, and artificial neural networks—that have been used to anticipate agricultural output. We also discuss the data sources, preprocessing techniques, and remote sensing data that are used to anticipate agricultural yields.

Conclusion: The ability of machine learning approaches to anticipate agricultural yields has shown considerable promise, and by enhancing crop management techniques, they have the potential to revolutionize agriculture. But there are still a lot of problems to be solved, such the requirement for high-quality data, the interpretability of the models, and the scalability of the algorithms. In the future, greater focus should be placed on creating more accurate and efficient machine learning (ML) models as well as gathering more diversified and high-quality data to increase the predictive accuracy.

[2] "Crop yield prediction using deep learning: A survey," Computers and Electronics in Agriculture, vol. 175, 2020

Authors: Z. Liu, Y. Dong, and Y. Li

Abstract: Deep learning has recently been applied to assess agricultural production, and several studies have shown that it outperforms traditional machine learning methods in this regard. In this survey, we present a comprehensive overview of deep learning-based agricultural production prediction. We begin by outlining the essential deep learning theories and principles. Finally, a summary is given of the datasets and data preparation techniques used in deep learning-based agricultural yield prediction. The most common deep learning models, including fully connected neural networks, convolutional neural networks, recurrent neural networks, and their modifications, are discussed after that. We also discuss the challenges associated with predicting agricultural productivity using deep learning as well as the evaluation metrics. Finally, we discuss some future directions for agricultural production prediction using deep learning.

Conclusion: Deep learning has surpassed more traditional machine learning algorithms in crop production prediction, which has shown to have a lot of promise. Predicting agricultural production using deep learning still has a number of challenges, such as a lack of large datasets, a high cost of compute, and a difficult time understanding the results. More attention should be paid to developing deeper learning models that are more efficient and intelligible as well as to acquiring high-quality data in order to boost the precision of the predictions.

[3] Crop yield prediction using machine learning: A comprehensive study," Precision Agriculture, vol. 22, no. 2, pp. 257-284, 2021.

Authors: S. Bera, M. Maity, and D. K. Maiti

Abstract: Predicting crop yields is a crucial problem in precision agriculture, and machine learning (ML) techniques have demonstrated considerable promise in this area because to their capacity to analyze massive amounts of data and produce precise forecasts. We provide a thorough analysis of current studies on agricultural production prediction using ML approaches in this work. We go over the several machine learning (ML) techniques that have been applied to forecast agricultural yields, such as decision trees, random forests, support vector machines, and artificial neural networks. We also go through the information sources and preprocessing methods used to forecast crop yields, such as weather, soil, and remote sensing data. Furthermore, we use a variety of assessment measures to examine the efficacy of several ML models for agricultural yield prediction. In conclusion, we emphasize the difficulties and potential paths for future study in agricultural production prediction using ML approach.

Conclusion: Machine learning techniques have demonstrated considerable promise in agricultural production prediction, and through enhancing crop management approaches, they have the potential to revolutionize agriculture. But there are still a lot of problems to be solved, such the requirement for high-quality data, the interpretability of the models, and the scalability of the algorithms. In the future, greater focus should be placed on creating more effective and accurate machine learning (ML) models as well as gathering more diversified and high-quality data to increase the predictive accuracy.

[4] A review on crop yield prediction using machine learning," International Journal of Agricultural and Biological Engineering, vol. 14, no. 4, pp. 25-34, 2021.

Authors: N. Y. Basak, A. Pal, and S. Roy

Abstract: In agriculture, forecasting crop yields is essential because it helps farmers to make informed crop management decisions. Machine learning (ML) techniques have been used to anticipate agricultural productivity because of their ability to examine vast amounts of data and provide accurate projections. In this review, we present a summary of the most recent research on predicting agricultural productivity using ML techniques. We discuss the many machine learning (ML) approaches—including decision trees, random forests, support vector machines, and artificial neural networks—that have been used to anticipate agricultural output. We also discuss the data sources, preprocessing techniques, and remote sensing data that are used to anticipate agricultural yields. Finally, we discuss the challenges and potential directions for further research. Finally, we highlight the challenges and future research directions in crop yield prediction using ML techniques.

Conclusion: Machine learning techniques have shown great promise in predicting crop yields, and they have the potential to revolutionize agriculture by improving crop management practices. However, there are still many challenges to be addressed, such as the need for high-quality data, the interpretability of the models, and the scalability of the algorithms. In the future, more attention should be paid to developing more efficient and accurate ML models, as well as collecting more diverse and high-quality data to improve the accuracy of the predictions.

[5] A Crop yield prediction using machine learning algorithms: A review," Journal of the Indian Society of Agricultural Statistics, vol. 76, no. 1, pp. 38-52, 2022.

Authors: R. Patel, P. Kumar, and A. Kumar.

Abstract: Crop yield prediction is a crucial task in agriculture for decision making related to crop management. Machine learning (ML) algorithms have emerged as a popular technique for crop yield prediction because of their ability to analyze large datasets and provide accurate predictions. This study presents a comprehensive review of the literature on crop yield prediction using ML algorithms. The review discusses various ML algorithms, data sources, and preprocessing techniques used for crop yield prediction. It also highlights the performance evaluation metrics used for assessing the accuracy of the predictions. Finally, the study concludes with future research directions and challenges in crop yield prediction using ML algorithms.

Conclusion: Machine learning algorithms have shown promising results in predicting crop yields and can aid in decision making related to crop management. However, there are still several challenges to be addressed, such as data quality, interpretability of models, and scalability of algorithms. In the future, more efforts should be made towards developing efficient and accurate models, collecting high-quality data, and improving the interpretability and scalability of the models.

[6] A Crop yield prediction using machine learning techniques: A systematic review," Agricultural and Forest Meteorology, vol. 311, pp. 107986, 2022.

Authors: D. L. Nogueira, J. N. R. Silva, and A. F. L. Cunha

Abstract: Predicting crop output is a crucial component of agriculture, and the use of machine learning (ML) techniques has increased recently. This systematic review extensively examines the literature on agricultural output prediction using machine learning methodologies, paying close attention to the various ML algorithms, data sources, and assessment measures used. The review covers publications published between 2010 and 2021 and includes research from various global regions. The study's conclusion includes a discussion of the challenges and potential opportunities in predicting agricultural productivity using ML techniques.

Conclusion: Machine learning techniques have shown great potential in predicting crop yields, and they can help farmers make informed decisions regarding crop management. However, several challenges still need to be addressed, such as data quality, interpretability of models, and scalability of algorithms. In the future, more attention should be paid to developing efficient and accurate models, collecting high-quality data, and improving the interpretability and scalability of the models.

CHAPTER 3

PROBLEM FORMULATION

3.1 PROBLEM IDENTIFICATION

Crops prediction using machine learning techniques is a powerful tool that can revolutionize the agricultural industry by providing accurate forecasts of crop yields. Traditional crop yield prediction methods frequently rely on manual observations, historical data, and expert knowledge. These methods are time-consuming, subjective, and incapable of reliably forecasting agricultural yields under variable environmental conditions. Farmers experience difficulties making educated judgements about planting schedules, fertilizer application, irrigation, and other agronomic practices as a result.

The goal of this project is to develop a predictive model using machine learning techniques that can accurately forecast crop yields based on various factors such as nitrogen, phosphorus, potassium, temperature, rainfall, and pH. By leveraging historical data and analyzing the relationships between these variables and crop productivity, we aim to create a reliable and efficient system that can assist farmers in optimizing their agricultural practices and maximizing crop yields.

The problem statement highlights the limitations of existing methods for crop yield prediction and emphasizes the need for a more advanced and accurate approach. The proposed solution using ML techniques aims to address these challenges and provide farmers with a tool that can support data-driven decision-making for improved crop management and increased agricultural Productivity.

3.2 PROPOSED SYSTEM

The proposed system uses a variety of machine learning algorithms, such as decision trees, XGBoost, random forests, support vector machines (SVM), naive Bayes, and logistic regression, and incorporates various factors, such as nitrogen, phosphorus, potassium, temperature, rainfall, and pH. Crop prediction is a critical activity in agriculture since it aids farmers in selecting crops, allocating resources, and optimizing yields. This technology seeks to give farmers precise predictions and insightful information by fusing ML algorithms with crucial agricultural metrics. The strategy considers critical soil nutrients including nitrogen, phosphorus, and potassium, all of which are crucial for crop development and output. By examining historical data and contrasting it with crop yields, machine learning (ML) algorithms including decision trees, XGBoost, random forests, SVM, naive Bayes, and logistic regression may successfully identify patterns and connections between nutrient levels and crop performance.

The approach considers environmental parameters such as temperature, rainfall, and pH in addition to soil nutrients. These factors have a significant impact on crop growth and development. The system may learn from previous data and find hidden correlations and trends between various environmental conditions and agricultural yields by utilizing the capabilities of ML algorithms. This enables accurate predictions and insights into how temperature, rainfall, and pH levels impact crop production.

The chosen ML algorithms—decision tree, XGBoost, random forest, SVM, naive Bayes, and logistic regression—each have their unique strengths. Decision tree algorithms offer interpretability, making it easier to understand the decision-making process. XGBoost and random forest algorithms are known for their ability to handle complex relationships and improve prediction accuracy. SVM algorithms are effective in handling both linear and non-linear data. Naive Bayes algorithms provide fast and efficient predictions, while logistic regression algorithms are well-suited for binary classification problems. By employing this comprehensive ML system, farmers can gain valuable insights into crop prediction and make data-driven decisions. This can lead to optimized resource allocation, improved crop yields, and increased overall agricultural productivity. Ultimately, this integrated approach aims to enhance the efficiency and sustainability of crop production while enabling farmers to adapt to changing environmental conditions and make informed choices for their agriculture.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behavior of the system.

Benefits of functional requirements:

- Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application
- A functional requirement document helps you to define the functionality of a system or one of its subsystems.
- Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
- Errors caught in the Functional requirement gathering stage are the cheapest to fix. Support user goals, tasks, or activities

4.2 SYSTEM REQUIREMENTS SPECIFICATION

The System requirement specification for crops prediction using ml techniques is a document that outlines the technical and functional requirements of the system. It includes a comprehensive description of the system's features, functions, and capabilities, as well as any limitations, constraints, or dependencies that must be considered during the development process. Overall, the system requirement specification for a crops prediction using ml techniques should provide a clear and detailed description of the system's features, functions, and capabilities, as well as any technical and functional requirements that must be met for the system to perform its intended tasks effectively. This document serves as a foundation for the development and testing of the system and helps to ensure that it meets the needs and expectations of its users.

4.2.1 SOFTWARE REQUIREMENTS

VS CODE:

Visual Studio Code (VS Code) is a robust integrated development environment (IDE) that may considerably enhance the development workflow for image and video-based bird identification applications. It includes a fast code editor with features like syntax highlighting, code completion, and debugging to assist developers in writing clean, error-free code. Furthermore, VS Code includes a robust extension ecosystem that allows developers to incorporate new features and integrations into their software development workflow. Extensions for several computer languages, frameworks, and libraries are available, including those required for image and video-based bird recognition. Furthermore, VS Code can interact with testing frameworks, allowing developers to build and execute automated tests to check that the product is functioning properly.

Fig 1 VS CODE

PYTHON:

Python, a popular computer language, has been used to recognize and identify birds in various applications in ornithology. Ornithologists and researchers create bird detection models using Python's many libraries and tools for image processing, computer vision, and machine learning. To identify bird species in photos and videos, these models employ methods such as object detection and classification. Furthermore, Python's user-friendly syntax and extensive community support make it a good choice for both beginners and specialists in ornithology. With the growing demand for bird monitoring and conservation activities, the usage of Python for bird detection has become critical in offering useful insights into the lives of birds and their environments

Fig 2 PYTHON

GOOGLE COLAB

Developers can write and execute Python code in a browser-based environment using Google Colab, also referred to as Google Colaboratory, which is a free cloud-based service offered by Google. It offers developers a practical approach to collaborate on tasks and work on machine learning projects without worrying about hardware restrictions. Google Colab can be a helpful tool for creating and refining machine learning models in the context of bird image recognition initiatives. It gives users access to sophisticated hardware that can accelerate the training of machine learning models, such as GPUs and TPUs.

Fig 3 google colab

OPERATING SYSTEM – WINDOWS 7/HIGHER:

Numerous data analysis and natural language processing (NLP) applications are compatible with Windows, a commonly used operating system. The majority of the necessary programs and libraries, including Python, NLTK, Genism, and sci-kit-learn, are supported by Windows.

Windows includes a vast ecosystem of software programs, including code editors, productivity tools, and integrated development environments (IDEs). Use popular IDEs like Visual Studio Code, PyCharm, or Anaconda, which are fully compatible with Windows and offer a seamless development experience, for subject modeling on regional tweets.

Windows offers a user-friendly graphical user interface that makes it simple to navigate and operate. When using tools and programs that need interaction and visualization, like data exploration, data preprocessing, and result analysis, this can be advantageous.

Fig 4 Operating System

FLASK

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. It offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Fig 5 Flask

4.2.2 HARDWARE REQUIREMENTS

To support a software system or application, a computer's CPU, memory, storage, and input/output devices are examples of particular physical components known as hardware requirements. The ability of the software to operate successfully and efficiently on the computer or device where it will be installed or used is ensured by this crucial part of software development.

PROCESSOR (INTEL i3 OR HIGHER):

The type of processor used in a chatbot system can vary depending on the specific requirements of the application, the expected workload, and the available hardware options. In general, a chatbot can be run on a range of processors, from low-end processors like Intel Atom or Celeron to higher-end processors like Intel Core i5 or i7, or even dedicated server processors like Intel Xeon.

The processor should have a sufficient number of cores and threads to support the software's workload, and it should be capable of executing instructions quickly to ensure that the chatbot can respond to user inputs in real-time. Additionally, the processor should have a suitable clock speed and cache size to ensure optimal performance.

Overall, the specific processor used in a chatbot system will depend on the particular requirements of the application, the expected workload, and the available hardware options.

An entry-level Intel Core series processor, or Intel i3, is suitable for simple computing tasks like word processing, web browsing, and light multimedia usage. However, a more powerful processor, such as an Intel i5, i7, or i9, would be advised for more demanding tasks like video editing, gaming, or running multiple applications at once.

The most advanced processors can easily tackle more demanding tasks since they have more cores, faster clock rates, and larger cache memory. Your individual needs and use requirements will ultimately determine the best CPU for you.

Fig 6 Processor

RAM (4GB OR HIGHER):

RAM (Random Access Memory) in a chatbot system refers to the memory that the system uses to temporarily store data and instructions that the processor needs to access quickly. In a chatbot system, the RAM is used to store the chatbot's working memory, which includes the conversation history, user context, and any other information needed to process user requests and generate responses. The amount of RAM required for a chatbot system can vary depending on the complexity of the application, the expected volume of user interactions, and the size of the data that needs to be stored in memory. Generally, a chatbot system would require at least 2-4 GB of RAM to function efficiently. However, more complex chatbots that handle large amounts of data or require advanced machine learning algorithms may require significantly more RAM to perform effectively. Having enough RAM in a chatbot system is important as it helps to ensure that the chatbot can quickly process user requests and generate responses in real-time. Without sufficient RAM, the chatbot may experience delays, freeze, or crash, resulting in a poor user experience. For the majority of contemporary operating systems and common computing operations like web browsing, word editing, and video playback, 4GB of RAM is regarded as the minimum requirement. You can run more programs and applications at once without encountering slowdowns or crashes the more Memory you have. A bigger RAM capacity can boost the speed and efficiency of the entire system, especially while multitasking or using memory- demanding software.

Fig 7 RAM

CHAPTER 5

SYSTEM DESIGN

5.1 BLOCK DIAGRAM

The proposed approach will recommend the best crop based on a few soil-related parameters.

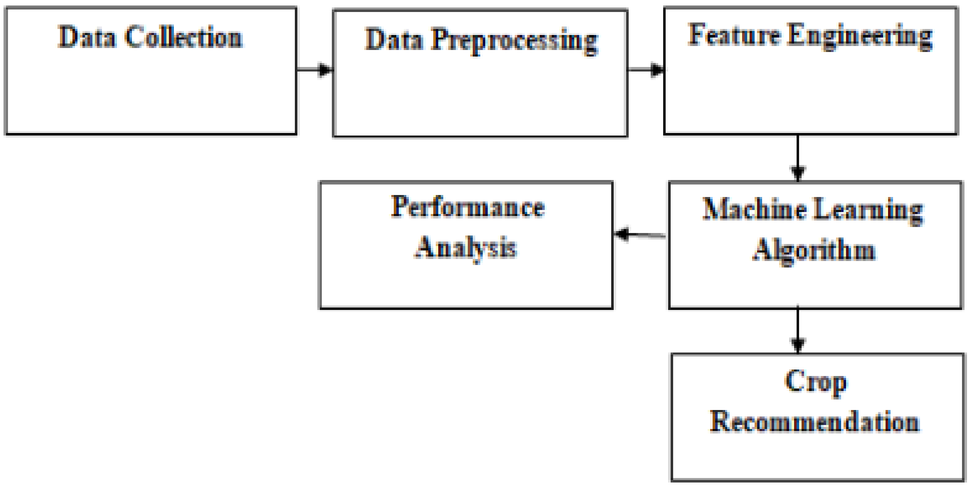


Fig 8 methodology for crop recommendation

The methodology of the proposed system consists of many components.

Data Collection:

Data collection is the process of compiling and examining data from various sources. Data collecting allows one to maintain track of prior events so that data analysis may be used to find repeating patterns. The dataset uses 7 characteristics and 22 distinct crops as class labels. (i) The ratio of nitrogen (N); (ii) the ratio of phosphorus (P); (iii) The soil's potassium content ratio (K), (iv) Temperature in degrees Celsius, (v) The percentage of relative humidity, (vi) the pH level, and (vii) the amount of rainfall in millimetres.

Data Pre-Processing:

After collection of data from different sources, the next step is to pre-processed it before the model can be trained. Data preprocessing is the act of transforming raw data into a format that analysts and data scientists may utilize in machine learning algorithms to discover insights or anticipate outcomes. Finding missing values is the data processing technique used in this study. It is challenging to obtain every data point for each record in a dataset. A lack of data may be indicated by empty cells, values like null, or a particular character like a question mark. There were no missing values in the dataset that was utilized for the research.

Feature Engineering:

Feature engineering derives features (characteristics, qualities, and attributes) from raw data using domain knowledge. The goal is to leverage these additional features to raise the calibre of ML output.

Training Set:

A data set with labelled data is referred to as a training set. Vectors for the input and output are both present. Using this dataset, the model is trained using supervised machine learning techniques.

Testing Set:

A data set without any tagged information is referred to as a testing set. With the use of the training data set, it makes a prediction about the result. Neither it nor the training data set are impacted.

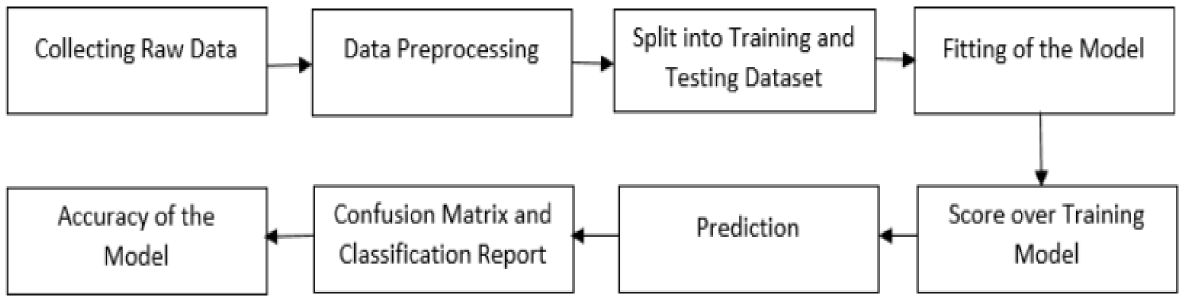


Fig 9 Proposed System

Using the train_test_split() function of the scikit learn module, the dataset is divided into a training dataset and a testing dataset. The dataset, which contains 2200 data, has been split into two datasets: a training dataset, which contains 1760 data, and a testing dataset, which contains 440 data.

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.

Checking the score over the training dataset:

Using a trained machine learning model, scoring, also known as prediction, is the process of generating values from fresh incoming data. Each model's score over a training dataset is calculated using the model.score() function, which reveals how well the model has learnt.

Predicting the model:

"Prediction" refers to the outcome of an algorithm after it has been trained on a prior dataset and applied to fresh data when predicting the likelihood of a particular result. utilizing the test feature dataset and the predict() function to predict the model. The output was provided as an array of forecasted numbers.

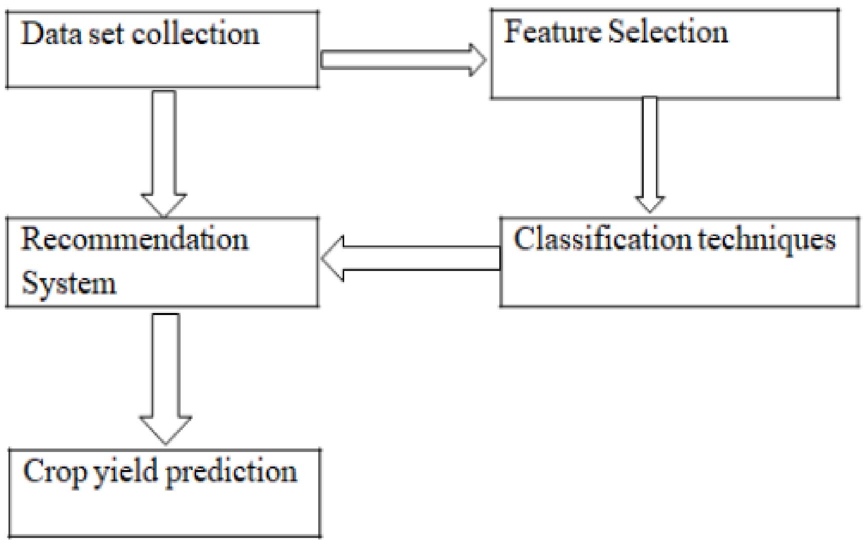


Fig 10 Block Diagram for recommendation

Confusion Matrix and Classification Report:

The metrics module in the scikit learn package was used to import the confusion matrix and classification report algorithms, which use test datasets actual labels and projected values to produce their results.

Confusion Matrix:

The frequency of true negatives, false negatives, true positives, and false positives is given by the confusion matrix.

Classification Report:

A classification algorithm's predictions are measured using the Classification Report metric. Precision, recall, and the model's f1 score are provided.

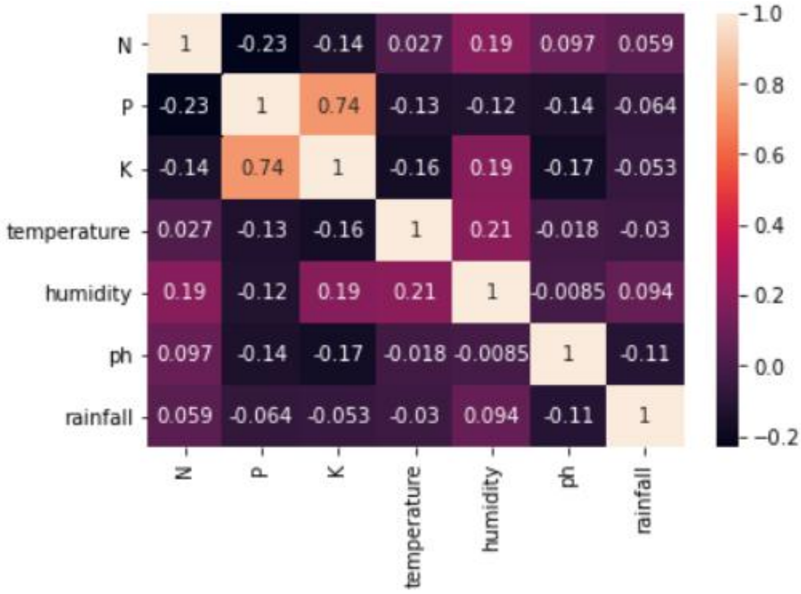


Fig11 correlation

Performance Analysis:

Performance analysis is a specialist field that use systemic goals to enhance performance and decision-making.

Crop Recommendation:

The model will suggest the best crop to grow on the specified soil based on the N P K, temperature, humidity, and ph.

Precision is the capacity of a classifier to count the number of positive predictions that are generally accurate. The ratio of true positives to the total of both true and false positives for each class is used to compute it.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Where Prediction Accuracy Is Precision-Positive

Recall: A classifier's recall is its capacity to extract all positive examples from the confusion matrix. The ratio of true positives to the total of true positives and false negatives for each class is used to compute it.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Where Recall: The proportion of accurately detected positives

F1 score: with 0.0 being the poorest and 1.0 being the greatest, is a weighted harmonic mean of recall and accuracy. F1 scores are frequently lower than accuracy assessments since precision and recall are taken into account during the computation**.**

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Accuracy:

The proportion of accurate forecasts to total predictions in the number of correct guesses.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

TP -True Positive	TN- true negatives
FN-false Negatives	FP False positives

CHAPTER 6

ALGORITHMS

6.1 DECISION TREE

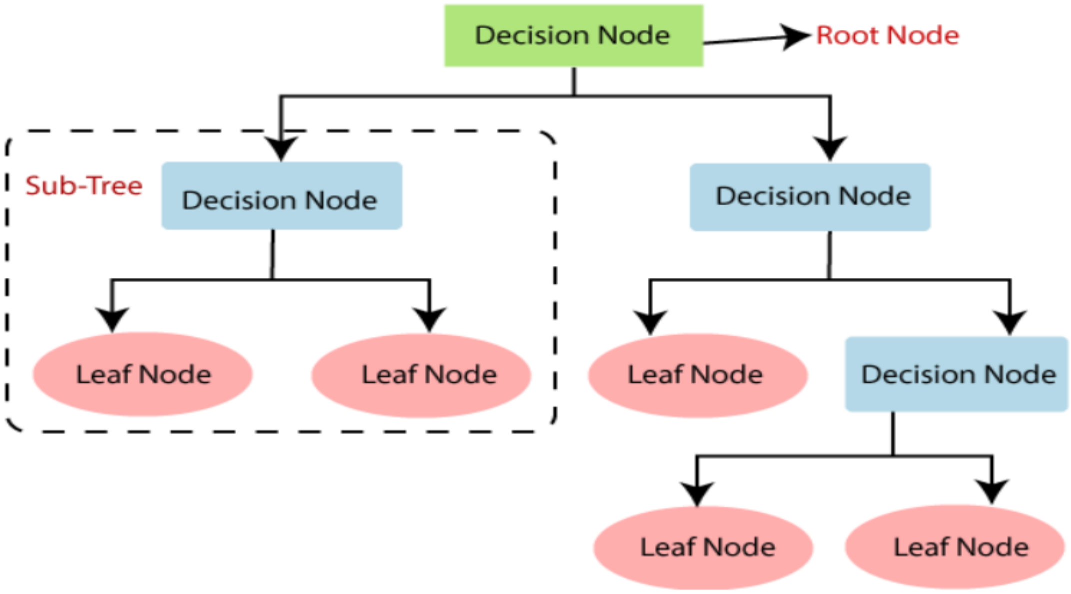


Fig 12 Decision Tree

Due of its simplicity and interpretability, the Decision Tree method is commonly employed in crop prediction. It generates a hierarchical structure of decision nodes depending on factors such as soil type, temperature, and rainfall. Each leaf node reflects a forecast crop type or yield. Decision Trees are good at processing categorical and numerical data, making them suited for agricultural datasets. However, decision trees are prone to overfitting, which can be addressed using ensemble approaches such as Random Forest. Decision Trees give insights into the decision-making process, allowing farmers to understand the elements impacting crop results.

Entropy:

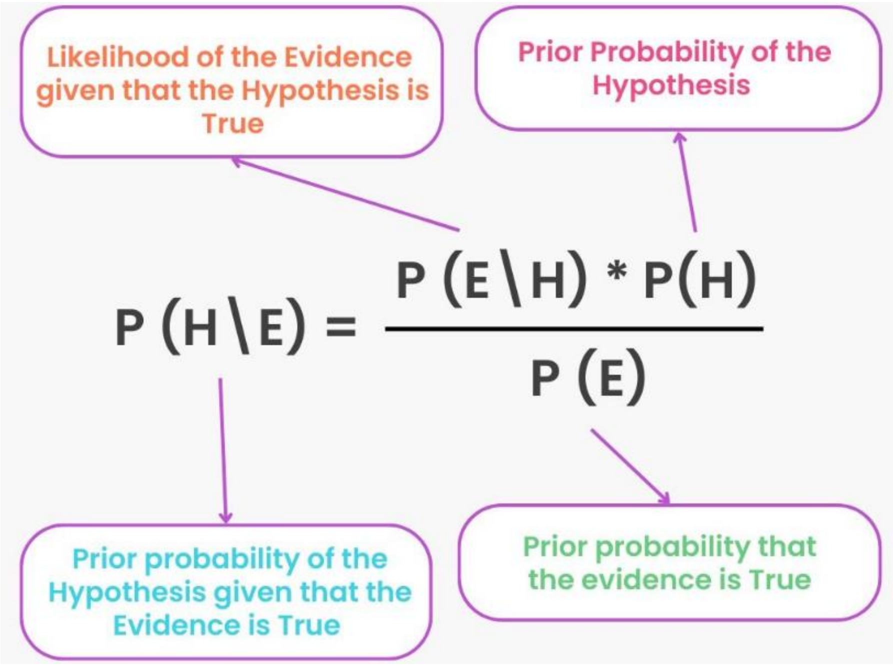
$$H(S) = -\sum Pi(S) \log2 Pi(S)$$

Information Gain:

$$IG(S,A) = H(S) - \sum_{v \in Values(A)} (|S_v|/S) H(S_v)$$

6.2 NAIVE BAYES

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.



Because of its simplicity and effectiveness, Naive Bayes is a probabilistic technique commonly employed in crop prediction. It is computationally efficient since it assumes that characteristics are conditionally independent given the class label. Naive Bayes evaluates the odds of distinct crop classes by analysing characteristics such as weather conditions, soil quality, and nutrient levels. It updates the probability depending on the observed data using Bayes' theorem. Naive Bayes can handle both categorical and numerical characteristics, and it works especially well with huge datasets. It is resistant to irrelevant characteristics and can make accurate predictions in real-time circumstances.

6.3 SUPPORT VECTOR MACHINE

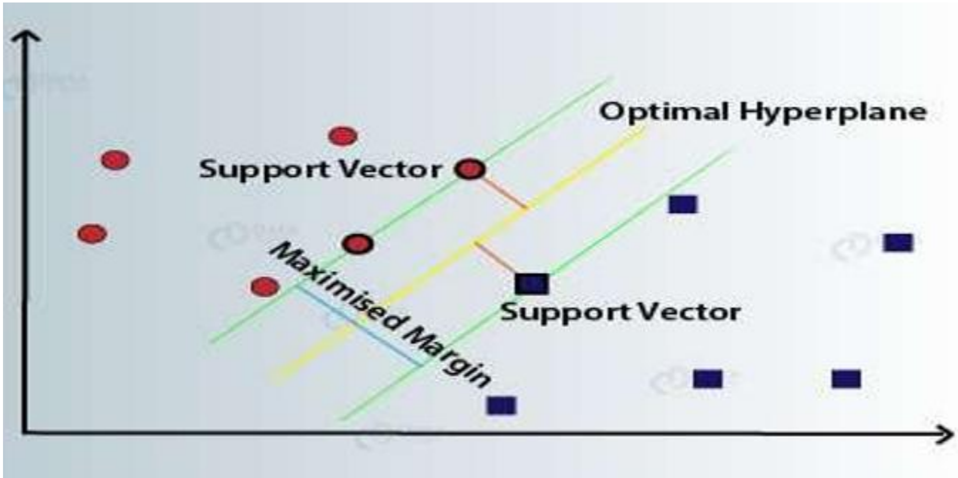
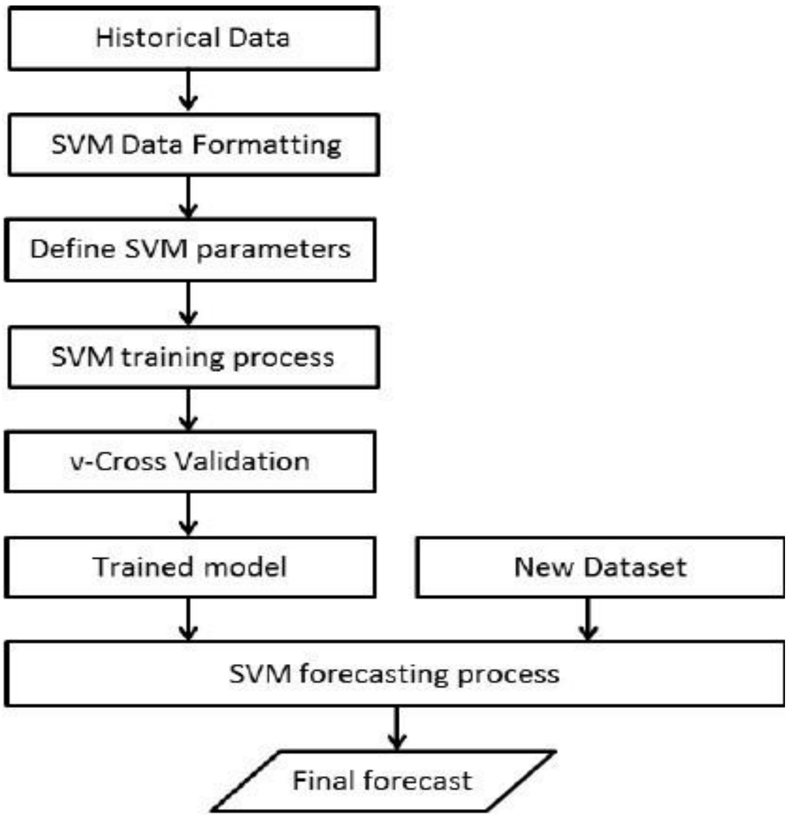


Fig 13 Support Vector Machine

A supervised machine learning method called the "Support Vector Machine" (SVM) may be applied to classification and regression problems. However, it is mostly used in categorizing issues. This method plots each data point as a point in an n-dimensional space, where n is the number of features, and each feature's value is represented by a specific coordinate. Then, classification is achieved by identifying the hyperplane that most effectively separates the two classes.



6.4 LOGISTIC REGRESSION

Logistic regression is a common Machine Learning method that belongs to the Supervised Learning approach. It is used to forecast the categorical dependent variable from a group of independent factors. A categorical dependent variable's output is predicted using logistic regression.

As a result, the conclusion must be categorical or discrete. It can be Yes or No, 0 or 1, true or False, and so on, but instead of presenting the precise values like 0 and 1, it presents the probability values that fall between 0 and 1. Except for how they are employed, Logistic Regression and Linear Regression are quite similar. Logistic regression is used to solve classification difficulties, whereas linear regression is used to solve regression problems.

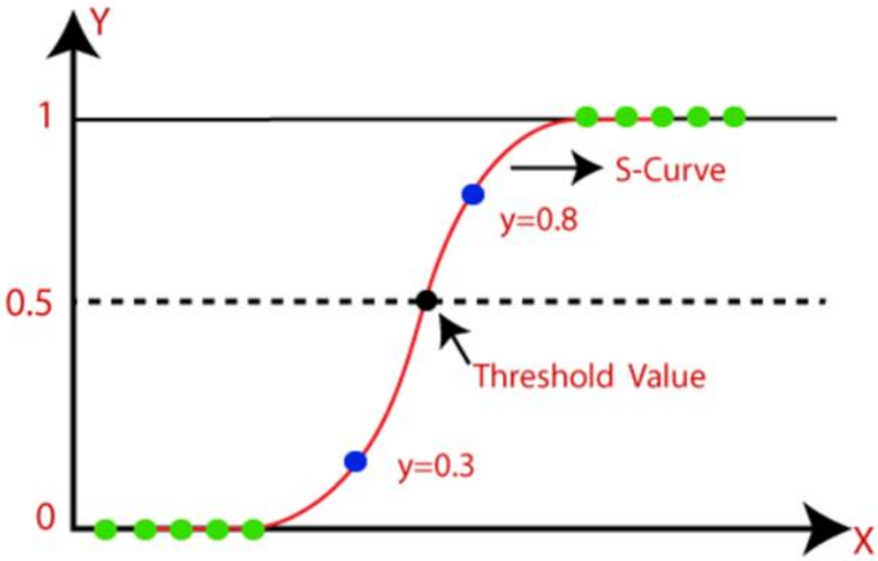


Fig 14 Logistic Regression

Logistic Regression is a common crop prediction technique, particularly for binary classification problems such as evaluating whether a crop will be healthy or unhealthy. It simulates the link between the input features and the likelihood of a specific result. When the goal variable is categorical and requires a probabilistic interpretation, Logistic Regression is appropriate. Logistic Regression can determine the likelihood of a certain crop state by analyzing characteristics such as soil quality, temperature, and insect presence. The model's coefficients give insight into the significance of each feature, assisting in understanding the influence of factors on crop health.

6.5 RANDOM FOREST

The supervised learning strategy uses the well-known machine learning algorithm Random Forest. Both classification and regression problems in machine learning may benefit from its use. It is based on the idea of ensemble learning, a technique that includes combining several classifiers to address a challenging problem and improve the performance of the model. A classifier called "Random Forest" "contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." The random forest gathers the predictions from each decision tree and predicts the ultimate result based on the majority vote of projections, as opposed to relying on a single decision tree.

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

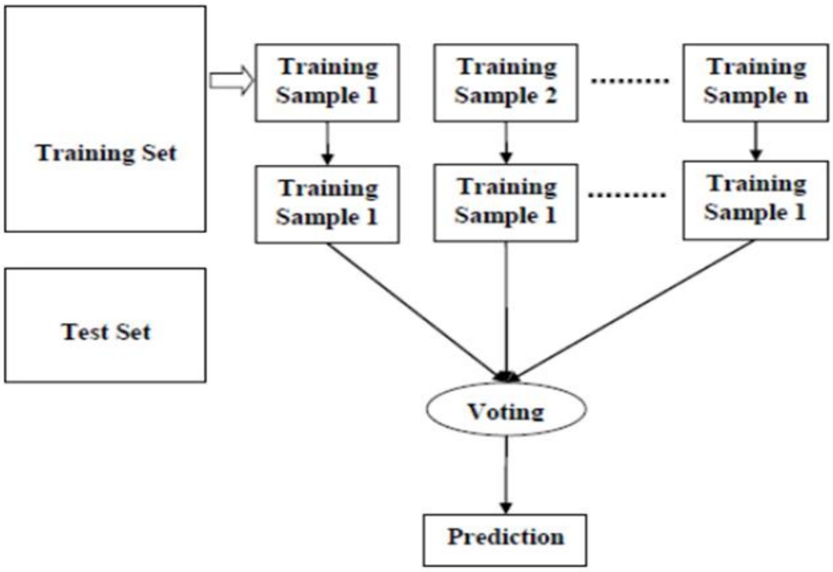


Fig 15 Random Forest

Random Forest is a sophisticated ensemble learning algorithm that may be used in agriculture to estimate crop yield. It makes accurate predictions by combining numerous decision trees. Random Forest may take into account a variety of factors in crop forecast, including soil type, meteorological conditions, temperature, and rainfall. The program can discover patterns and correlations between these characteristics and crop productivity by training on historical data. Random Forest's randomization aids in reducing overfitting and improving generalization. It also gives feature significance, which enables farmers to determine the major aspects influencing crop output. Because Random Forest is resistant to outliers and missing data, it is appropriate for real-world agricultural datasets.

6.6 XG BOOST

XGBoost (Extreme Gradient Boosting) is a highly effective gradient boosting method that is commonly used in crop prediction. To produce accurate predictions, it integrates the predictions of numerous weak learners (decision trees). XGBoost optimizes a loss function by repeatedly adding trees to the ensemble and learning from prior trees' faults. It is capable of handling complicated interactions between attributes and of capturing non-linear correlations in crop data. XGBoost assigns priority to features, helping farmers to find the most relevant aspects for crop prediction. It is computationally efficient and capable of dealing with big datasets, making it appropriate for agricultural applications.

Gradient Boosting: XGBoost is a gradient boosting solution that successively combines weak prediction models (often decision trees) to generate a strong predictive model. Each new model is taught to fix the errors committed by the older models, increasing the total forecast accuracy.

Customizable Loss Functions: XGBoost enables users to create their own unique loss functions. This feature is helpful when working with particular problem domains or when the conventional loss functions do not adequately address the issue. It offers flexibility for model optimization based on particular assessment standards.

Tree Pruning: To regulate the depth of each individual decision tree in the ensemble, XGBoost uses a method known as tree pruning. Pruning aids in lowering overfitting and enhancing the algorithm's capacity for generalization.

Missing Values: During the training phase, XGBoost includes built-in ability to manage missing values in the dataset. Assigning missing values to the left or right child nodes in each tree, it automatically learns the optimum path for them to go in. Cross-Validation: The k-fold cross-validation method is supported by XGBoost and enables you to assess the model's performance on various subsets of the training data. This makes it easier to gauge the model's generalizability and efficiently adjust the hyperparameters.
Overall, XGBoost's popularity has grown as a result of its efficacy, scalability, and efficiency in a variety of machine learning situations. Each node in the tree has seen effective application in a variety of fields, including banking, healthcare, natural language processing, and computer vision.

CHAPTER 7

IMPLEMENTATION

One problem we pointed up before is that only one factor—weather or soil—is taken into account when predicting crop growth suitability. However, we think that for the greatest and most precise forecast, each of these factors should be taken into account simultaneously. This is due to the fact that, even if a particular soil type may be ideal for supporting a particular crop, the yield may suffer if the local weather is unfavorable to that crop. We provide a crop recommendation system that anticipates crop suitability by taking into account all pertinent information, such as temperature, rainfall, and soil nutrients, in order to solve the restrictions. This system's main focus is on performing its primary function, which is to use algorithms to suggest crops to farmers. The user has an easy and reliable way to choose and plan the crop thanks to the parameters. to create a solid model that can precisely predict crop sustainability in each state depending on soil nutrients and environmental conditions.

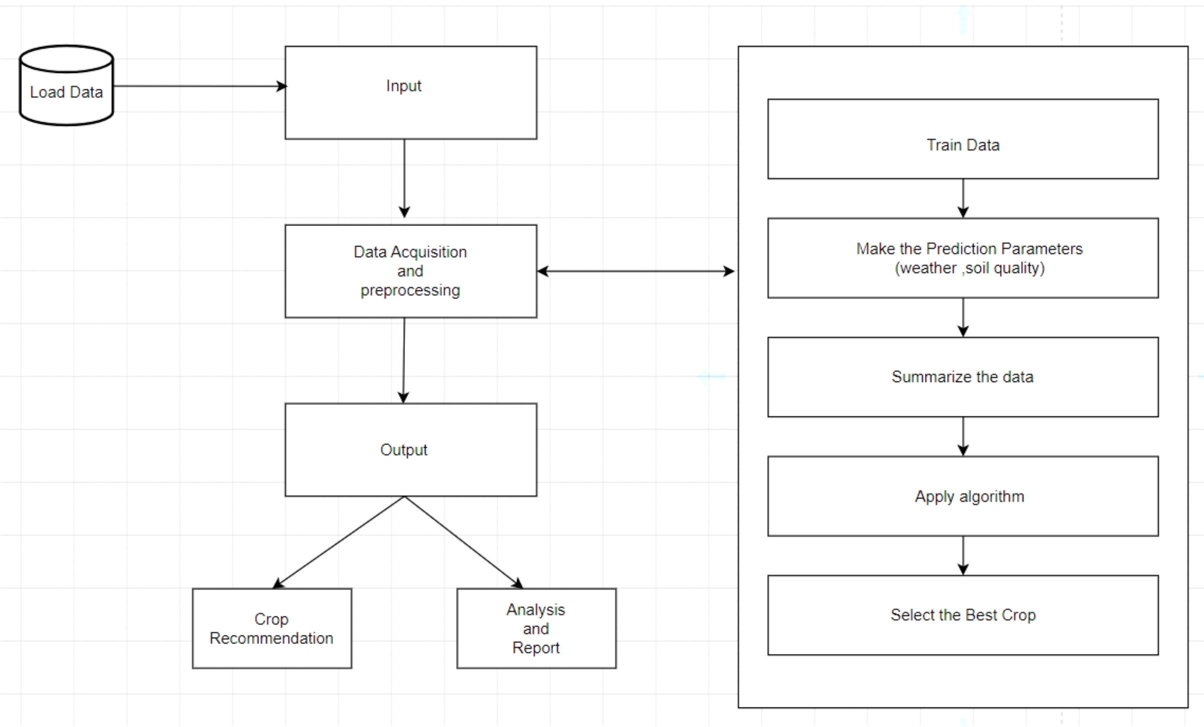
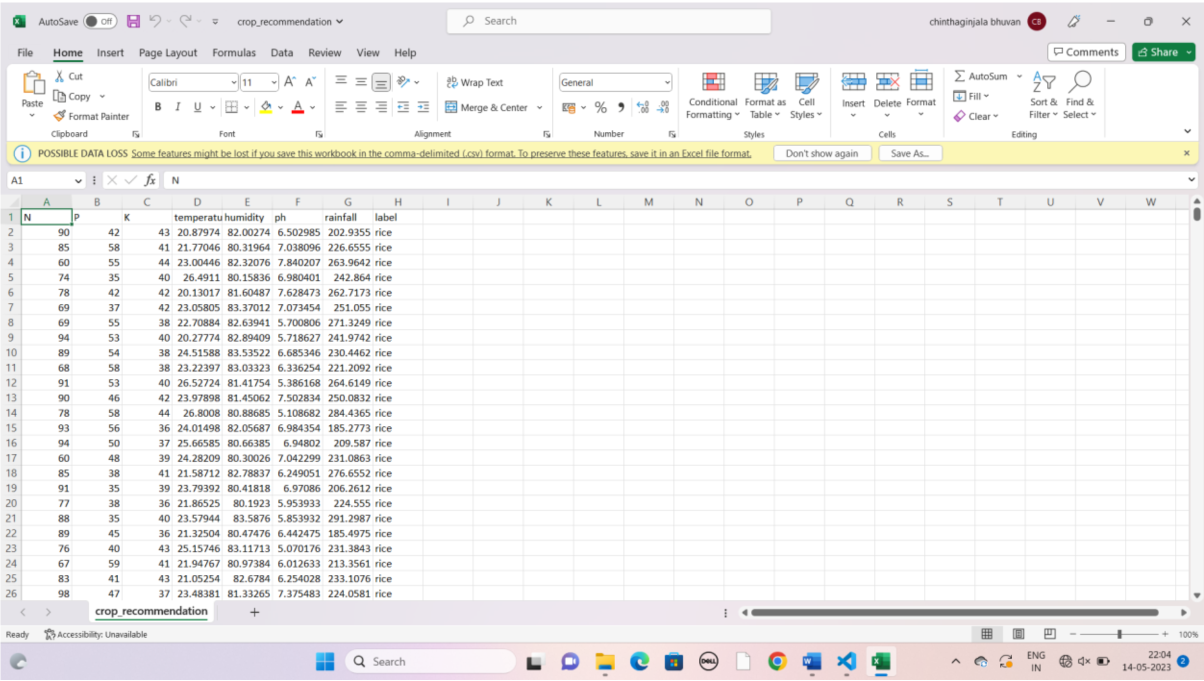


Fig 16 Data flow diagram

One of the initial steps we perform during deployment is an examination of the data. We did this to check for relationships between the different dataset characteristics. Any machine learning method's accuracy is determined by the quantity of parameters and the quality of the training dataset. Numerous studies in this field have used environmental factors to forecast agricultural sustainability; some have focused primarily on yield, while others have only considered economic factors.

In order to provide the farmer with an exact and reliable recommendation on which crop would be best for his land, we combined climate factors like rainfall, temperature, and soil ph with soil parameters like soil nutrients. Using the read_csv() method from the pandas package, we import the dataset.



	N	P	K	temperatu	humidity	ph	rainfall	label
2	90	42	43	20.87974	82.00274	6.502985	202.9355	rice
3	85	58	41	21.77046	80.31964	7.038096	226.6555	rice
4	60	55	44	23.00446	82.32076	7.840207	263.9642	rice
5	74	35	40	26.4911	80.15836	6.980401	242.864	rice
6	78	42	42	20.13017	81.60487	7.628473	262.7173	rice
7	69	37	42	23.05805	83.37012	7.073454	251.055	rice
8	69	55	38	22.70884	82.63941	5.700806	271.3249	rice
9	94	53	40	20.27774	82.89409	5.718627	241.9742	rice
10	89	54	38	24.51588	83.53522	6.685346	230.4462	rice
11	68	58	38	23.22397	83.03323	6.336254	221.2092	rice
12	91	53	40	26.52724	81.41754	5.386168	264.6149	rice
13	90	46	42	23.97898	81.45062	7.502834	250.0832	rice
14	78	58	44	26.8008	80.88685	5.108682	284.4365	rice
15	93	56	36	24.01498	82.05687	6.984354	185.2773	rice
16	94	50	37	25.66585	80.66385	6.94802	209.587	rice
17	60	48	39	24.28209	80.30026	7.042299	231.0863	rice
18	85	38	41	21.58712	82.78837	6.249051	276.6552	rice
19	91	35	39	23.79392	80.41818	6.97086	206.2612	rice
20	77	38	36	21.86525	80.1923	5.953933	224.555	rice
21	88	35	40	23.57944	83.5876	5.853932	291.2987	rice
22	89	45	36	21.32504	80.47476	6.442475	185.4975	rice
23	76	40	43	25.15746	83.11713	5.070176	231.3843	rice
24	67	59	41	21.94767	80.97384	6.012633	213.3561	rice
25	83	41	43	21.05254	82.6784	6.254028	233.1076	rice
26	98	47	37	23.48381	81.33265	7.375483	224.0581	rice

Fig 17 Data set

Data Preprocessing:

Sometimes, real-world data has noise, missing values, and is in an improper format that prevents it from being easily incorporated into machine learning models. To clean data and make it acceptable for a machine learning model, which increases the model's efficacy and accuracy, data preparation is an essential activity.

Data cleaning and preparation for use in machine learning algorithms make data preprocessing a key step. Preprocessing is largely concerned with correcting any missing data as well as eliminating any outliers or inaccurate data. There are two techniques to fill in any gaps in the data. The first choice is to remove the whole row that contains the inaccurate or missing data. Although this strategy is straightforward to apply, it works best with big datasets.When used to small datasets, this technique can significantly reduce the size of the dataset, especially when there are a lot of missing values. The accuracy of the outcome may be significantly impacted by this. Because our dataset is so little, we won't employ this tactic.

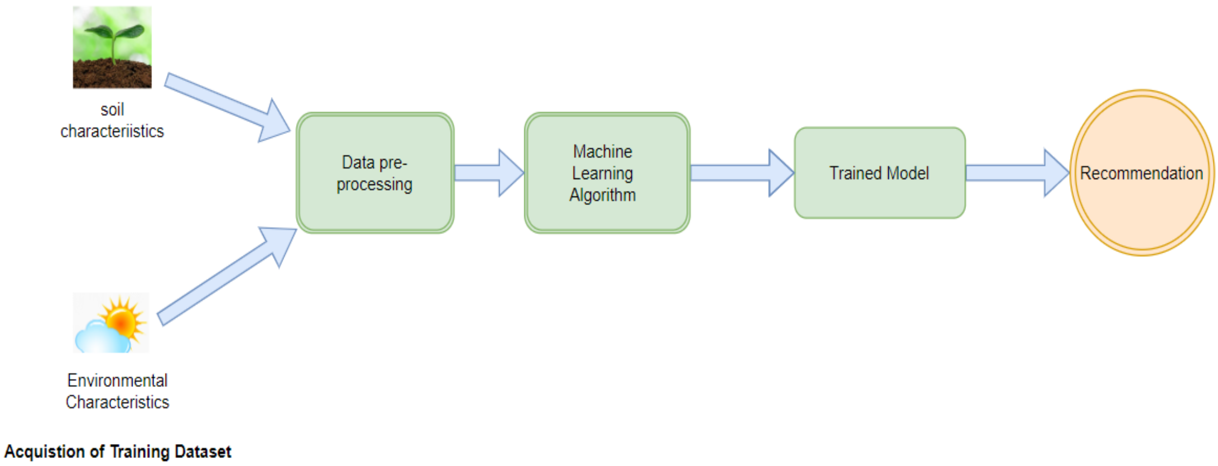
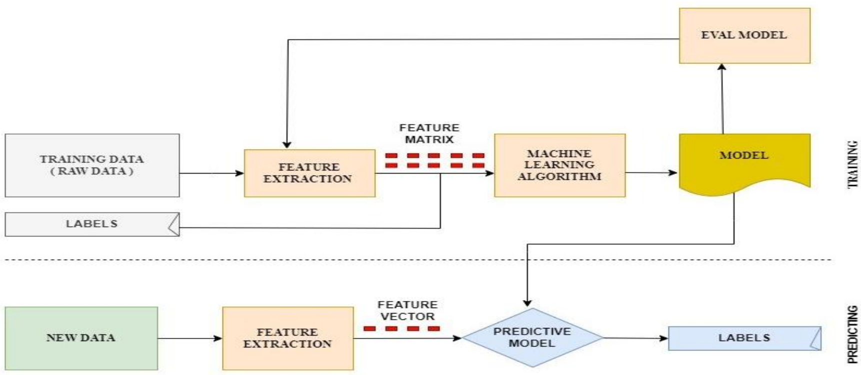


Fig.18 Data preprocessing

Training and Testing data:

We used many machine learning strategies to obtain reliable findings because the suggested model has to be trained and evaluated under a range of circumstances. Here, the data has been educated to anticipate the crop that can be produced depending on a variety of elements, such as soil nutrients and environmental conditions. We train the data to forecast the precise crop that will be grown using a variety of input factors. We create predictions based on the X test data and fit the data to the X, Y training values. We ran 100 training samples on the model. The best model is one that has the lowest loss, and this model is utilized for testing and assessment.



SAMPLE CODE:

```
# Importing libraries
from __future__ import print_function
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('../Data-processed/crop-recommendation.csv')
df.head()
df.tail()
df.shape
df.columns
df['label'].unique()
df.dtypes
df['label'].value_counts()
sns.heatmap(df.corr(),annot=True)
#sepearating features and target label
features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = df['label']
#features = df[['temperature', 'humidity', 'ph', 'rainfall']]
labels = df['label']
# Initialzing empty lists to append all model's name and corresponding name
acc = []
model = []
# Splitting into train and test data
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size =
0.2,random_state =2)
```

#decision tree

```
from sklearn.tree import DecisionTreeClassifier
DecisionTree =
DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)
DecisionTree.fit(Xtrain,Ytrain)
predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)
print(classification_report(Ytest,predicted_values))
from sklearn.model_selection import cross_val_score
# Cross validation score (Decision Tree)
score = cross_val_score(DecisionTree, features, target,cv=5)
#saving traines decision tree model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
DT_pkl_filename = '../models/DecisionTree.pkl'
# Open the file to save as pkl file
DT_Model_pkl = open(DT_pkl_filename, 'wb')
pickle.dump(DecisionTree, DT_Model_pkl)
# Close the pickle instances
DT_Model_pkl.close()
```

#Guassian Naïve Bayes

```
from sklearn.naive_bayes import GaussianNB
NaiveBayes = GaussianNB()
NaiveBayes.fit(Xtrain,Ytrain)
predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
```

```
# Cross validation score (NaiveBayes)
score = cross_val_score(NaiveBayes,features,target,cv=5)
score
#saving trained guassian Naïve Bayes model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
NB_pkl_filename = '../models/NBClassifier.pkl'
# Open the file to save as pkl file
NB_Model_pkl = open(NB_pkl_filename, 'wb')
pickle.dump(NaiveBayes, NB_Model_pkl)
# Close the pickle instances
NB_Model_pkl.close()
```

#Support Vector Machine(svm)

```
from sklearn.svm import SVC
# data normalization with sklearn
from sklearn.preprocessing import MinMaxScaler
# fit scaler on training data
norm = MinMaxScaler().fit(Xtrain)
X_train_norm = norm.transform(Xtrain)
# transform testing dataabs
X_test_norm = norm.transform(Xtest)
SVM = SVC(kernel='poly', degree=3, C=1)
SVM.fit(X_train_norm,Ytrain)
predicted_values = SVM.predict(X_test_norm)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
# Cross validation score (SVM)
score = cross_val_score(SVM,features,target,cv=5)
score
#Saving trained SVM model
```

```
import pickle
# Dump the trained SVM classifier with Pickle
SVM_pkl_filename = '../models/SVMClassifier.pkl'
# Open the file to save as pkl file
SVM_Model_pkl = open(SVM_pkl_filename, 'wb')
pickle.dump(SVM, SVM_Model_pkl)
# Close the pickle instances
SVM_Model_pkl.close()
```

#Logistic Regression

```
from sklearn.linear_model import LogisticRegression
LogReg = LogisticRegression(random_state=2)
LogReg.fit(Xtrain,Ytrain)
predicted_values = LogReg.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
# Cross validation score (Logistic Regression)
score = cross_val_score(LogReg,features,target,cv=5)
score
#saving trained Logistic Regression model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
LR_pkl_filename = '../models/LogisticRegression.pkl'
# Open the file to save as pkl file
LR_Model_pkl = open(DT_pkl_filename, 'wb')
pickle.dump(LogReg, LR_Model_pkl)
# Close the pickle instances
LR_Model_pkl.close()
```

#Random Forest

```
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)
print(classification_report(Ytest,predicted_values))
# Cross validation score (Random Forest)
score = cross_val_score(RF,features,target,cv=5)
score
#saving trained random forest model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
RF_pkl_filename = '../models/RandomForest.pkl'
# Open the file to save as pkl file
RF_Model_pkl = open(RF_pkl_filename, 'wb')
pickle.dump(RF, RF_Model_pkl)
# Close the pickle instances
RF_Model_pkl.close()
```

#XG boost

```
import xgboost as xgb
XB = xgb.XGBClassifier()
XB.fit(Xtrain,Ytrain)
predicted_values = XB.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('XGBoost')
print("XGBoost's Accuracy is: ", x)
print(classification_report(Ytest,predicted_values))
# Cross validation score (XGBoost)
```

```
score = cross_val_score(XB,features,target,cv=5)
score
#Saving Trained XGBoost model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
XB_pkl_filename = '../models/XGBoost.pkl'
# Open the file to save as pkl file
XB_Model_pkl = open(XB_pkl_filename, 'wb')
pickle.dump(XB, XB_Model_pkl)
# Close the pickle instances
XB_Model_pkl.close()
#Accuracy Comparison
plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
accuracy_models = dict(zip(model, acc))
for k, v in accuracy_models.items():
print (k, '-->', v)
```

CHAPTER 8

SNAPSHOTS

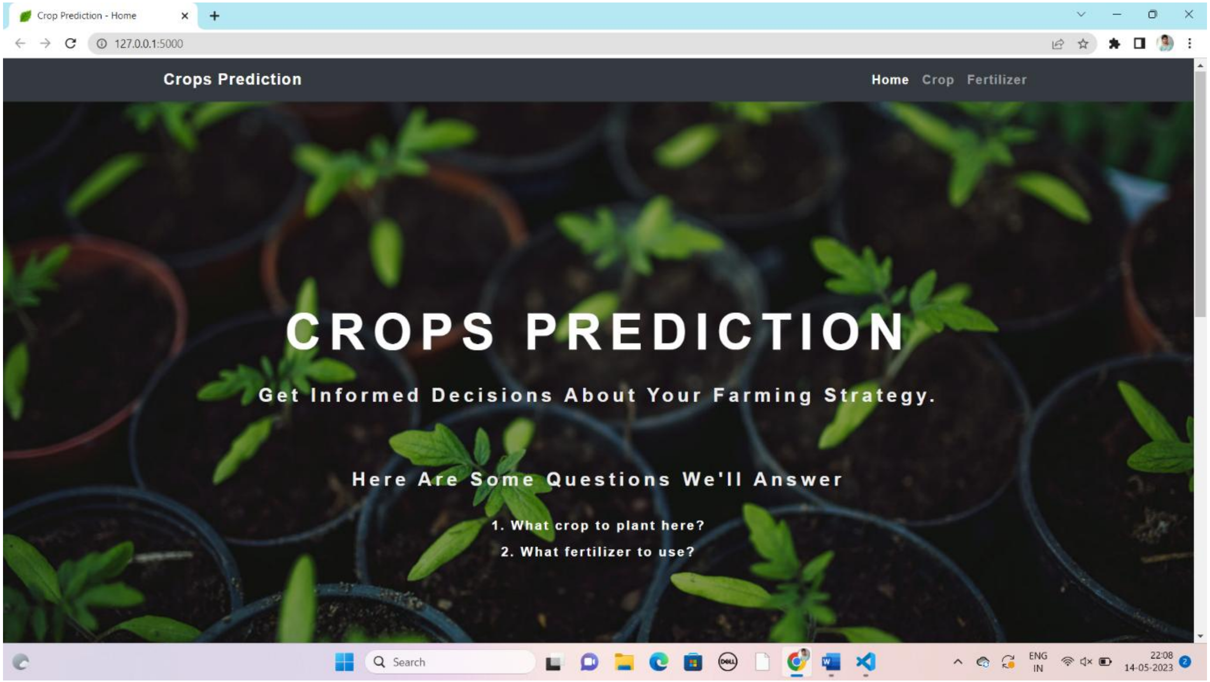


Fig 19 Home

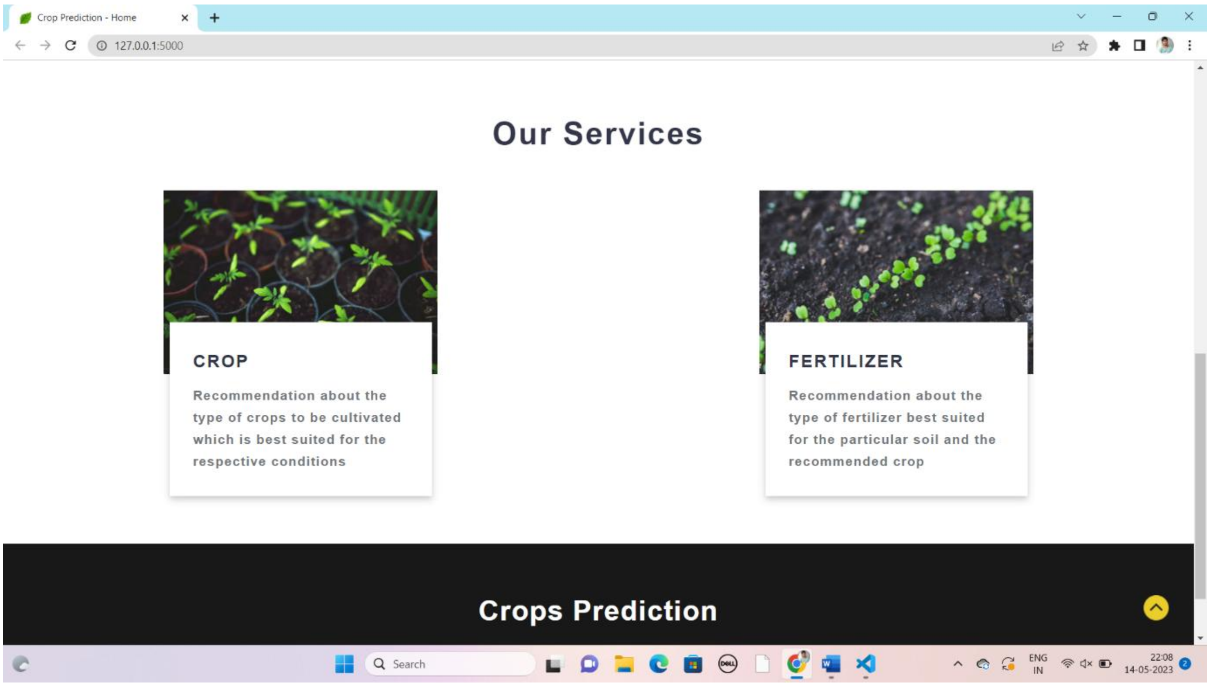


Fig 20 About

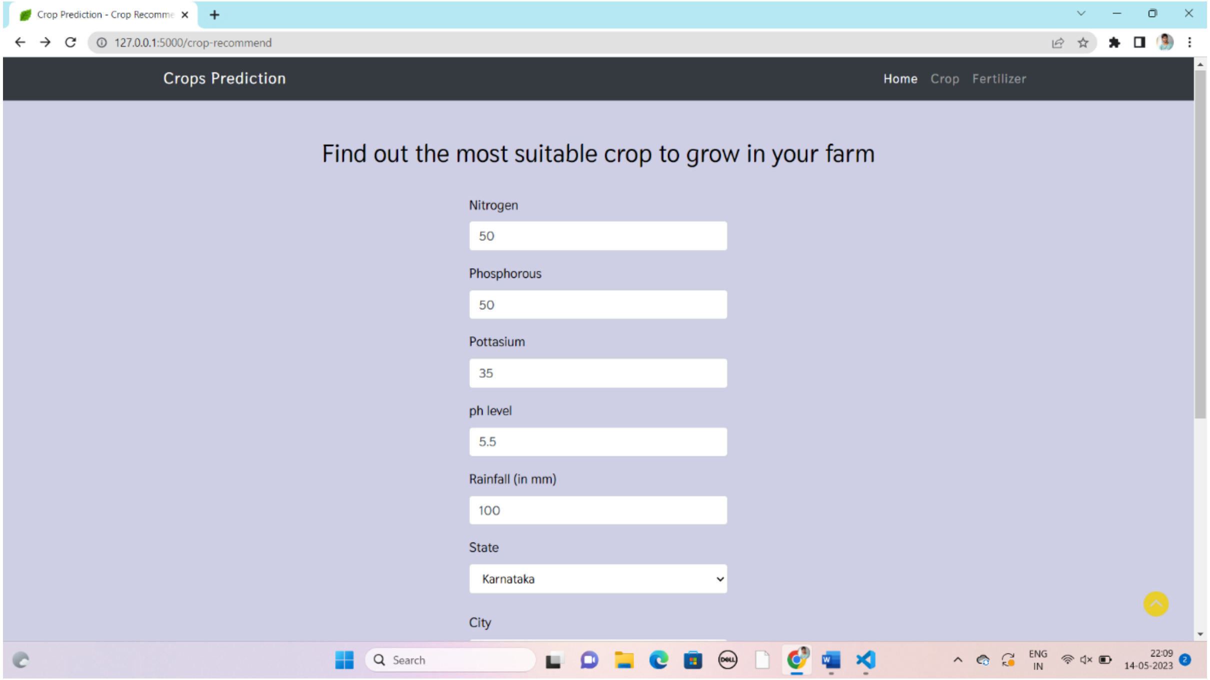


Fig 21 Crop

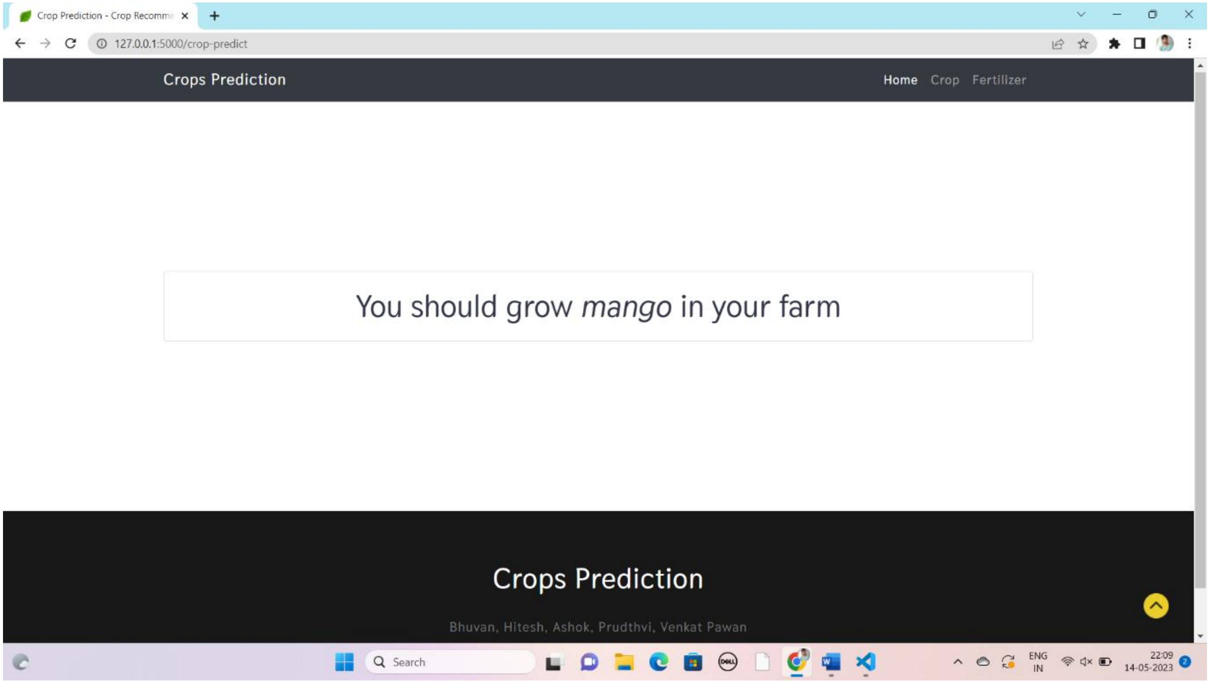


Fig 22 Crop result

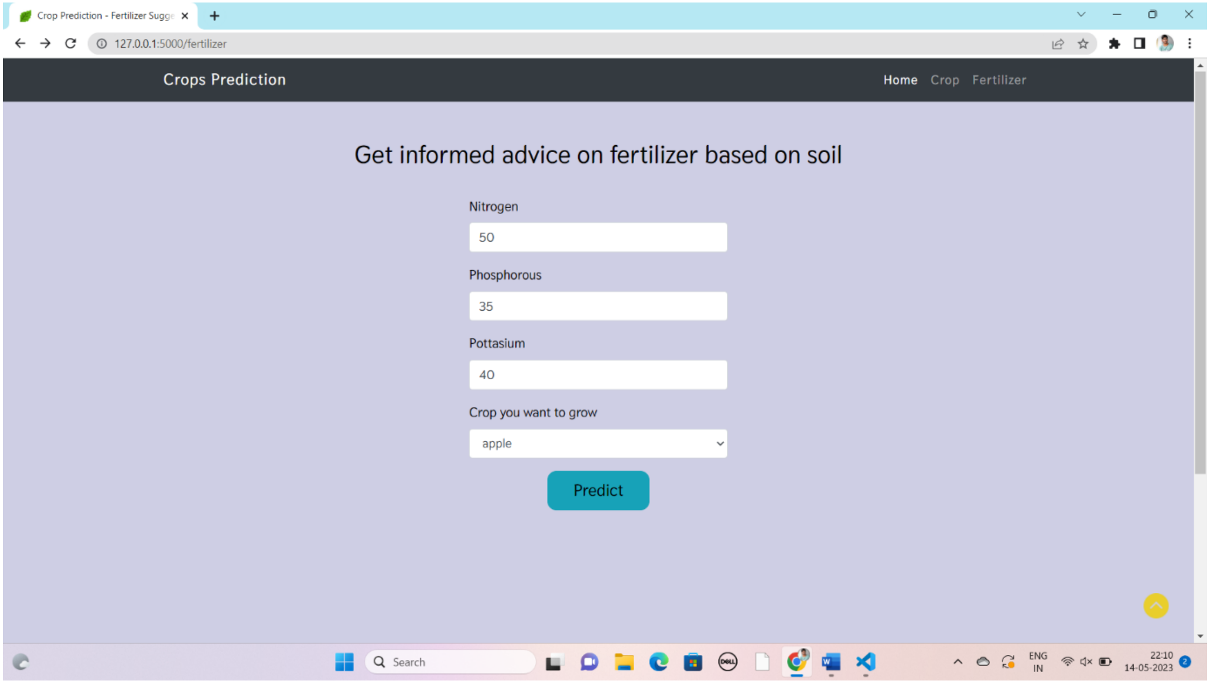


Fig 23 Fertilizer

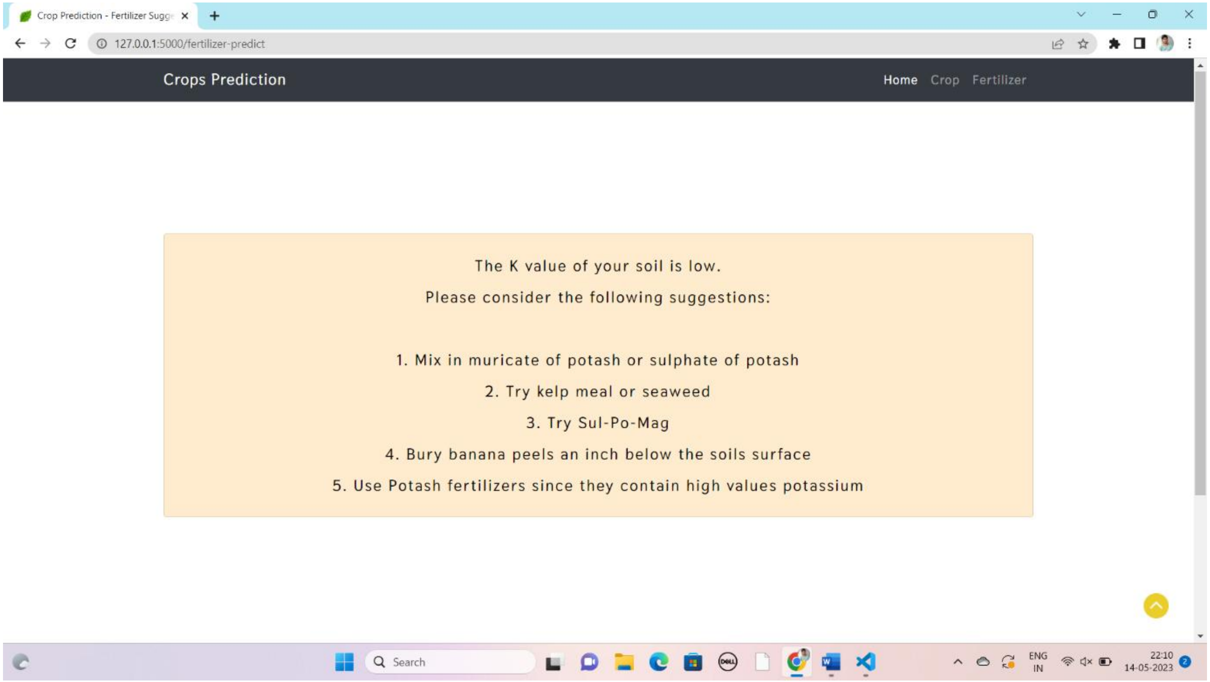


Fig 24 Fertilizer result

SUMMARY

The user-friendly crop prediction interface makes it simple for farmers to enter the necessary information. The interface has specific fields where users may enter the soil's nitrogen (N), phosphorus (P), and potassium (K) levels. It also has input sections for the soil pH, humidity, temperature, and rainfall. The interface provides a thorough evaluation of the environmental conditions required for crop growth by taking these important aspects into account.

The interface uses machine learning methods to process the data once the user enters the necessary information in order to suggest a crop. On a dataset of historical crop yields according to various environmental circumstances, the algorithms have been trained. This enables the interface to produce precise predictions depending on the particular set of parameters that the user has supplied.

The interface prominently shows the suggested crop along with other details like the typical yield and any particular conditions for successful growth. Farmers may use this information to organize their farming activities and make well-informed decisions.

The crop prediction interface based on environmental factors provides farmers with a practical and effective way to make well-informed crop selection decisions. The interface creates precise suggestions that are suited to the particular circumstances of the user's land by taking into account variables including nutrient levels, temperature, humidity, rainfall, and pH values. The interface gives farmers the power to select the best crops, maximize yields, and implement sustainable agricultural practices thanks to its user-friendly design, visual representations, and feedback system.

CHAPTER 9

TESTING

Testing is the process of looking for any flaws or weaknesses in a piece of work. It offers a means of testing whether parts, sub-assemblies, assemblies, and/or a finished product perform properly. It is the process of testing software to make sure that it satisfies user expectations and meets requirements without failing in an unacceptable way. Different test types exist. Every test type responds to a certain testing requirement.

The following objectives are sought to be accomplished during testing:

- To reaffirm the project's high calibre.
- To identify and fix any leftover mistakes from earlier steps.
- To verify that the software is a viable fix for the original issue.
- To guarantee the system's operational dependability.

9.1 TYPES OF TESTING

i) Unit Testing:

Unit testing examines a single unit or a collection of related units. It is classified as white box testing. Because testing will be dependent on the completeness and correctness of the test specification, they must be subjected to quality and verification review.

ii) Integration Testing:

Integration testing is the process of combining several components to generate output. In addition, if software and hardware components are related, the interaction between them is tested during integration testing. It may fall under both white-box testing and black-box testing. We have applied a top-down strategy to validate high-level components of a system before design and implementation have been completed. Our development process started with high-level components and we worked down the component hierarchy.

iii) System Testing:

System testing ensures that the program continues to function in multiple environments (e.g., operating systems). System testing is carried out using the entire system implementation and environment. It is classified as black box testing.

iv) Manual Testing:

The process of manually testing software for detects is known as manual testing.

CHAPTER 10

RESULT AND DISCUSSION

10.1 RESULT

The machine learning techniques used yielded encouraging crop forecast outcomes. With a score of 0.993, XGBoost had the greatest accuracy of the algorithms that were evaluated, followed by Decision Tree (0.990), Naive Bayes (0.990), Random Forest (0.990), Support Vector Machine (0.971), and Logistic Regression (0.952).

The predictions of many weak models (decision trees) are combined using the ensemble learning method XGBoost to produce a strong predictive model. Its remarkable success can be due to its resilience against overfitting, as well as its capacity to manage intricate connections between input data and output classes. The comparable accuracy levels of the Decision Tree, Random Forest, and Naive Bayes algorithms show how well they forecast crops. To make judgements based on feature splits, decision tree models construct a flowchart structure like a tree. A group of decision trees called Random Forest uses the collective knowledge of many different models to increase the precision of predictions. On the other hand, Naive Bayes uses probabilistic techniques based on Bayes' theorem to categorize the data.

The accuracy rates for Support Vector Machine (SVM) and Logistic Regression were 0.971 and 0.952, respectively. SVM develops an ideal hyperplane to divide the classes, whereas Logistic Regression models the association between the input characteristics and the likelihood of a certain crop class.

DECISION TREE:

The Decision Tree method is commonly employed in crop prediction. It generates a hierarchical structure of decision nodes depending on factors such as soil type, temperature, and rainfall. Each leaf node reflects a forecast crop type or yield. Decision tree got an accuracy of 0.9.

DecisionTrees's Accuracy is: 90.0

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.59	1.00	0.74	16
chickpea	1.00	1.00	1.00	21
coconut	0.91	1.00	0.95	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.74	0.93	0.83	28
kidneybeans	0.00	0.00	0.00	14
lentil	0.68	1.00	0.81	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	0.00	0.00	0.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	0.84	0.91	19
pigeonpeas	0.62	1.00	0.77	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.62	0.77	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.90	440
macro avg	0.84	0.88	0.85	440
weighted avg	0.86	0.90	0.87	440

NAIVE BAYES:

Because of its simplicity and effectiveness, Naive Bayes is a probabilistic technique commonly employed in crop prediction. It is computationally efficient since it assumes that characteristics are conditionally independent given the class label. Naive Bayes evaluates the odds of distinct crop classes by analyzing characteristics such as weather conditions, soil quality, and nutrient levels. Naive Bayes got an accuracy of 0.9909

```
Naive Bayes's Accuracy is:  0.990909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.88      1.00      0.93        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      1.00      1.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      1.00      1.00        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       1.00      0.75      0.86        16
  watermelon       1.00      1.00      1.00        15
...
    accuracy                           0.99       440
   macro avg       0.99      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440
```

SUPPORT VECTOR MACHINE:

SVM may learn from labelled data in the context of crop prediction to categories crops into distinct groups based on various traits or qualities. The data points that have the greatest impact on establishing the decision border between various crop classes are represented by the support vectors. These support vectors allow the SVM model generalize to effectively identify unseen data by serving as the borders between various crop groups. Support Vector Machine got an Accuracy of 0.9795.

```
SVM's Accuracy is:  0.9795454545454545
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      0.95      0.98        22
      cotton       0.95      1.00      0.98        20
      grapes       1.00      1.00      1.00        18
        jute       0.83      0.89      0.86        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      0.95      0.98        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      1.00      1.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      1.00      1.00        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       0.80      0.75      0.77        16
  watermelon       1.00      1.00      1.00        15
...
    accuracy                           0.98       440
   macro avg       0.98      0.98      0.98       440
weighted avg       0.98      0.98      0.98       440
```

LOGISTIC REGRESSION:

Logistic Regression is a common crop prediction technique, particularly for binary classification problems such as evaluating whether a crop will be healthy or unhealthy. It simulates the link between the input features and the likelihood of a specific result. When the goal variable is categorical and requires a probabilistic interpretation, Logistic Regression is appropriate. Logistic Regression can determine the likelihood of a certain crop state by analyzing characteristics such as soil quality, temperature, and insect presence. Logistic regression got an accuracy of 0.9522.

```
Logistic Regression's Accuracy is:  0.9522727272727273
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.86      0.75      0.80        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       0.86      0.90      0.88        20
      grapes       1.00      1.00      1.00        18
        jute       0.84      0.93      0.88        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       0.88      1.00      0.94        23
       maize       0.90      0.86      0.88        21
       mango       0.96      1.00      0.98        26
   mothbeans       0.84      0.84      0.84        19
    mungbean       1.00      0.96      0.98        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      0.95      0.97        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       0.85      0.69      0.76        16
  watermelon       1.00      1.00      1.00        15
...
    accuracy                           0.95       440
   macro avg       0.95      0.95      0.95       440
weighted avg       0.95      0.95      0.95       440
```

RANDOM FOREST:

Random Forest is a sophisticated ensemble learning algorithm that may be used in agriculture to estimate crop yield. It makes accurate predictions by combining numerous decision trees. Random Forest may take into account a variety of factors in crop forecast, including soil type, meteorological conditions, temperature, and rainfall. The program can discover patterns and correlations between these characteristics and crop productivity by training on historical data. Random forest got an accuracy of 0.9909.

```
RF's Accuracy is:  0.990909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.94      1.00      0.97        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.90      1.00      0.95        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      0.95      0.97        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      1.00      1.00        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       1.00      0.81      0.90        16
  watermelon       1.00      1.00      1.00        15
...
    accuracy                           0.99       440
   macro avg       0.99      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440
```

XG BOOST :

XGBoost (Extreme Gradient Boosting) is a highly effective gradient boosting method that is commonly used in crop prediction. To produce accurate predictions, it integrates the predictions of numerous weak learners (decision trees). XGBoost optimizes a loss function by repeatedly adding trees to the ensemble and learning from prior trees' faults. It is capable of handling complicated interactions between attributes and of capturing non-linear correlations in crop data. XGBoost assigns priority to features, helping farmers to find the most relevant aspects for crop prediction. XGBoost got an accuracy of 0.9931.

```
XGBoost's Accuracy is:  0.9931818181818182
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       0.96      1.00      0.98        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       1.00      0.93      0.96        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       0.96      1.00      0.98        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      0.95      0.97        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      1.00      1.00        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       0.94      1.00      0.97        16
...
    accuracy                           0.99       440
   macro avg       0.99      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440
```

ALGORITHM	ACCURACY
Decision Tree	0.9
Naive Bayes	0.9909
Support Vector Machine	0.9795
Logistic Regression	0.9522
Random Forest	0.9909
XG Boost	0.9931

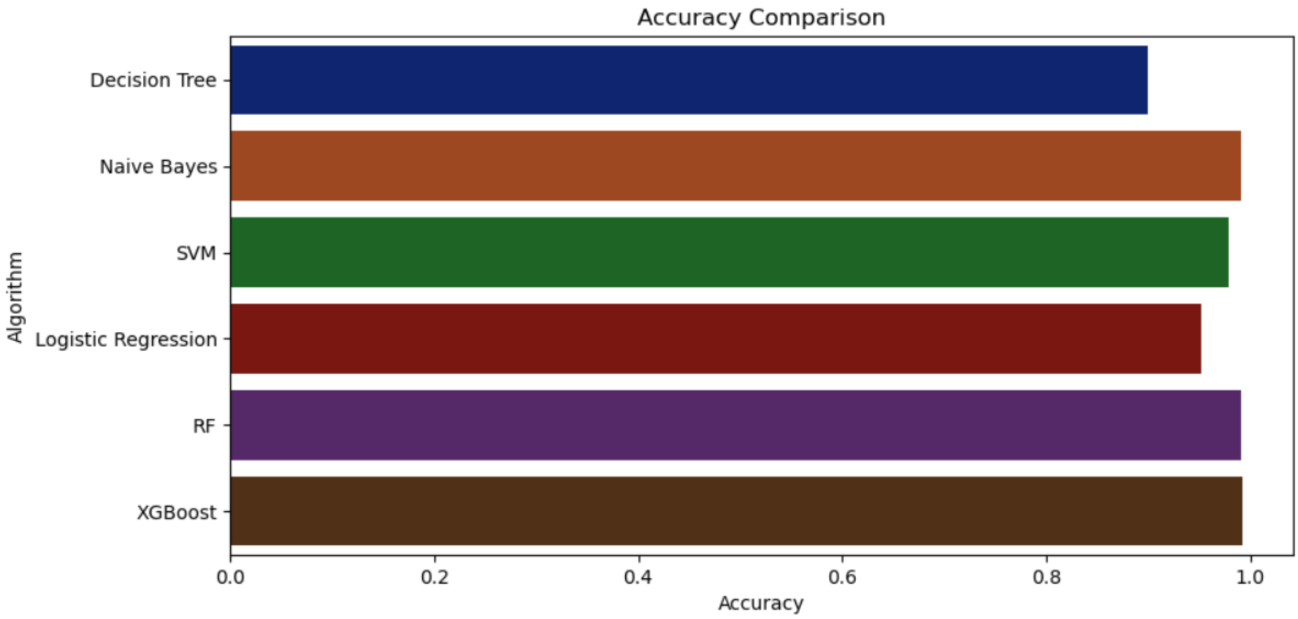


Fig 25 Accuracy comparison

In this project, we've created a user-friendly interface that considers crucial elements including pH levels, temperature, humidity, rainfall, and nutrient levels (N, P, and K). Farmers may get precise suggestions for the best crop to grow on their land by supplying these inputs.

In this model, soil characteristics and a crop database are employed. Using machine learning algorithms, the best crop is suggested for the particular soil. Of the techniques we examined, XG Boosting had the highest accuracy.

10.2 CONCLUSION

In conclusion, our study utilized machine learning to construct a crop forecast system. The implemented algorithms, such as XGBoost, Decision Tree, Naive Bayes, Random Forest, Support Vector Machine, and Logistic Regression, showed their prowess in accurately recommending suitable crops in response to input parameters like nutrient levels, temperature, rainfall, and pH values.

Because of its high accuracy, XGBoost has the potential to be a strong tool for crop prediction. Decision Tree, Random Forest, and Naive Bayes algorithms also performed well. The outcomes of this study can help farmers and agricultural professionals choose crops wisely, allocate resources efficiently, and boost agricultural output.

To improve the system's prediction powers, future study may investigate new machine learning methods, incorporate more varied information, and take temporal characteristics into account. The development of user-friendly interfaces and the incorporation of real-time data streams might also improve the crop prediction system's usability and applicability.

We all know that a lot of agricultural research has been done and is still being done to increase production, strengthen the Indian economy, and, most significantly, help farmers increase their income. The suggested approach would provide farmers with guidance on the optimum crop to grow on their property in order to achieve this. Consequently, farmers will benefit.

By giving suggestions on the best crop to grow on their property, the suggested method intends to assist farmers in increasing their productivity and revenue. This method has been created using agricultural research to help the Indian economy grow. The technology will offer farmers useful data they can use to choose crops wisely, leading to improved agricultural yields and increased profitability. By using this technology, farmers may take use of the advantages of machine learning approaches to optimize crop choices, leading to more fruitful harvests and more financial gain.

BIBLIOGRAPHY

1. "Crop yield prediction using machine learning: A review," Journal of Agricultural Science and Technology, vol. 23, no. 4, pp. 901-913, 2021. N Devi and V.K.Gupta
2. "Crop yield prediction using deep learning: A survey," Computers and Electronics in Agriculture, vol. 175, 2020. Z. Liu, Y. Dong, and Y. Li
3. Crop yield prediction using machine learning: A comprehensive study," Precision Agriculture, vol. 22, no. 2, pp. 257-284, 2021. S. Bera, M. Maity, and D. K. Maiti
4. A review on crop yield prediction using machine learning," International Journal of Agricultural and Biological Engineering, vol. 14, no. 4, pp. 25-34, 2021. N. Y. Basak, A. Pal, and S. Roy
5. A Crop yield prediction using machine learning algorithms: A review," Journal of the Indian Society of Agricultural Statistics, vol. 76, no. 1, pp. 38-52, 2022. R. Patel, P. Kumar, and A. Kumar.
6. A Crop yield prediction using machine learning techniques: A systematic review," Agricultural and Forest Meteorology, vol. 311, pp. 107986, 2022. D. L. Nogueira, J. N. R. Silva, and A. F. L. Cunha
7. D. D. M. Fonseca, D. D. de Medeiros, C. R. Dantas, and L. A. D. V. Barbosa, "A systematic review of crop yield prediction using machine learning: Progress and prospects," Information Processing in Agriculture, vol. 8, pp. 203-214, 2021.
8. J. Chen, H. Liu, Y. Zhang, L. Du, and W. Song, "Crop yield prediction based on remote sensing data and machine learning techniques: A review," Agricultural and Forest Meteorology, vol. 310-311, pp. 108279, 2022.
9. V. R. Velvizhi and R. Kumar, "Crop yield prediction using machine learning techniques: A review," Computers and Electronics in Agriculture, vol. 185, 106087, 2021.
10. Jain, S., Mahajan, S., & Sharma, A. (2020). Crop Yield Prediction using Machine Learning Algorithms: A Review. Journal of Computational and Theoretical Nanoscience, 17(10), 5098-5105.
11. J. Lee et al., "Crop yield prediction using deep learning: A review," Computers and Electronics in Agriculture, vol. 183, 2021.
12. S. K. Kumawat, S. K. Singh, and N. Kumar, "Crop yield prediction using machine learning: a review," SN Applied Sciences, vol. 2, no. 10, pp. 1-22, 2020