

CROP & FERTILIZER RECOMMENDATION USING LINEAR FOREST CLASSIFIER

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

DARVIN XAVIER P (Reg.No - 40110291)

KOUSHIK S (Reg.No - 40110641)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC I 12B Status by UGC I Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Darvin Xavier P (40110291)** and **Koushik S (40110641)** who carried out the Project entitled "**Crop & Fertilizer recommendation Using Linear Forest Classifier**" under my supervision from November 2023 to April 2024.

Internal Guide

Dr. T. JUDGI, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Jeppiaar Nagar, Raj. Gandhi Salai,
Chennai - 600 127.

Submitted for Viva voce Examination held on 10/4/24

Internal Examiner

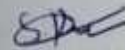
External Examiner

DECLARATION

I, **Koushik S (Reg.No - 40110641)**, here by declare that the Project Report entitled "**CROP AND FERTILIZER RECOMMENDATION USING LINEAR FOREST CLASSIFIER**" done by me under the guidance of **Dr. T. Judgi, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12/11/24

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. T. Judgi, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In Indian economy and employment agriculture plays major role. The most common problem faced by the Indian farmers is they do not opt crop based on the necessity of soil, as a result they face serious setback in productivity. This problem can be addressed through precision agriculture. This method takes three parameters into consideration, viz: soil characteristics, soil types and crop yield data collection based on these parameters suggesting the farmer suitable crop to be cultivated. Precision agriculture helps in reduction of non suitable crop which indeed increases productivity, apart from the following advantages like efficacy in input as well as output and better decision making for farming. This method gives solutions like proposing a recommendation system through AI to recommend suitable crop based on soil parameters with high specific accuracy and efficiency.

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES & LIST OF TABLES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
	2.1 Inferences from Literature Survey	10
	2.2 Open problems in Existing System	11
3	REQUIREMENT ANALYSIS	13
	3.1 Feasibility Study/Risk Analysis Of The Project	13
	3.2 Software Requirements Specification Document	15
4	DESCRIPTION OF PROPOSED SYSTEM	17
	4.1 Selected Methodology or process model	17
	4.2 Architecture / Overall Design of Proposed System	19
5	IMPLEMENTATION DETAILS	27
	5.1 Description of Software For Implementation and Testing Plan of The Proposed Model/System	27
	5.2 Project Management Plan	28
	5.3 Transition/ Software to Operations Plan	28
6	RESULTS AND DISCUSSION	30
	6.1 Result Output	30

6.2	Performance Analysis	32
7	SUMMARY AND CONCLUSIONS	35
7.1	Conclusion	35
7.2	Future Enhancement	35
	REFERENCES	38
	APPENDIX	38
	A.SOURCE CODE	38
	B.SCREENSHOTS	52
	C.RESEARCH PAPER	55
	D.CERTIFICATE	62

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.2.1	System Architecture	20
4.2.2	Data Flow Diagram (DFD)	21
6.1.1	Flask framework output	31
6.2.1	Random Forest Confusion Matrix	33
B.1	Home Screen	52
B.2	About Us	52
B.3	Our Services	53
B.4	Input Screen	53
B.5	Output Screen	54

LIST OF TABLES

TABLE NO	TABLE NAME	Page No.
2.1.1	Summary of Literature Survey	10
6.2.1	Comparitive Table	32

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
LR	Logistic Regression
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbors
DT	Decision Tree Classifier
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

Agriculture, as we all know, is the foundation of the Indian economy. Agriculture is an important occupation in India. More than 60% of the country's land is used for agriculture, which feeds 1.3 billion people. Agriculture is the cultivation of plants and animals. In India, agriculture gave rise to civilization. We need soil to cultivate crops. As a result, soil is a critical factor in agriculture. Soil health is essential for good food production. It provides the roots with essential nutrients, water, oxygen, and support. Soil is the foundation of the food system, as well as the location of all plants used in food production. In India, several soil varieties are available. They are alluvial soil (cotton, rice), black soil (sugarcane, sunflower), red soil (corn, ragi), laterite soil (pulses, tea, coffee), and so on. Many studies have been conducted to improve agricultural planning. The crop can be recommended using a machine learning technique.

India is one of the biggest producers of agricultural products and still has very less farm productivity. Productivity needs to be increased so that farmers can get more pay from the same piece of land with less labour . Precision agriculture provides a way to do it.

Precision farming, as the name implies, refers to the applying of precise and proper total of comment like fertilizers, soil etc. but in recent times the trend in agriculture has drastically evolved due to globalization various factors have affected the health of agriculture in India. Many new technologies have been evolved to regain the health.

Machine learning is an subfield of artificial intelligence that describes a machine's ability to mimic intelligent human behavior. Artificial intelligence systems are employed in the same way as humans do to automate complex tasks . Machine learning begins with data, such as financial transactions, individuals, or photos.

The information is collected and processed to be utilized as training data for the machine learning system. If the data is more then the software shows better results. After that, the developer select a ML model to use, input the data, and train the system to find patterns or make predictions on its own.

One such technique is precision agriculture, at the proper time to the crop for increasing its productivity and increasing its yields. Not all precision agriculture systems offer best results. precision agriculture is a technology of site-specific farming.

But in agriculture it is important that the recommendations made are accurate and precise because in case of errors it may lead to heavy material and capital loss. recommendation of crops is one major domain in precision agriculture.

Agriculture is one of the most important sectors in the world, providing food, fiber, and other essential resources for humanity. However, agriculture is also one of the most challenging sectors, facing a range of issues including climate change, resource scarcity, and a growing population that requires ever-increasing amounts of food.

To address these challenges, farmers need to make informed decisions about which crops to grow and how to manage them effectively. However, this is not always an easy task, as farmers need to take into account a range of factors, such as soil characteristics, weather conditions, and historical crop data, to make the best decisions.

This is where crop recommendation using machine learning (ML) techniques comes in. By leveraging advanced analytics and data-driven insights, farmers can make better decisions about crop selection and management, leading to more sustainable and profitable farming practices.

The process of crop recommendation using ML typically involves collecting data from a range of sources, including soil sensors, weather stations, historical crop data, and satellite imagery. This data is then fed into ML algorithms, which analyze the data and generate recommendations based on various factors, such as soil nutrient levels, precipitation, temperature, and other relevant variables.

The recommendations generated by the ML algorithms can help farmers to optimize their crop yields, reduce crop losses, and ultimately increase profitability. For example, if the algorithm determines that a particular crop is likely to perform well in a particular soil type and climate, it may recommend that the farmer plant that crop instead of another, less suitable crop.

1.1.AIM:

- In modern agriculture, maximizing and sustaining crop recommendation are the main objectives.
- One of the major problems constraining the development of an economically successful agriculture is nutrient deficiency for crop production.
- Farming aims to improve crop performance and environmental quality.

1.2.OBJECTIVES:

- ☐ To Predict the right crop which suits the soil
- ☐ To assist the farmer in growing the right crop by giving appropriate guidance.
- ☐ To streamline Agriculture in India.

CHAPTER 2

LITERATURE SURVEY

This proposed system has gone through various references and research over different journals and conference papers that are evolved for the past 10 years. The research and referrals of survey is conducted on different domains like AI, NLP,, Home environments that helped to developed the proposed system of AI based Control AAS. Those references and the detailed report on the references are listed as follows.

Priyadharshini A, Swapneel Chakraborty, et al.[1] (2021)proceedings of the fifth international conference on computing methodologies and communicationintelligent crop recommendation system using machine learning.

Agriculture plays a vital role in the socioeconomic fabric of India. Failure of farmers to decide on the best-suited crop for the land using traditional and non-scientific methods is a serious issue for a country where approximately 58 percent of the population is involved in farming. Sometimes farmers were failed to choose the right crops based on the soil conditions, sowing season, and geographical location. This results in suicide, quitting the agriculture field, moving towards urban areas for livelihood. To overcome this issue, this research work has proposed a system to assist the farmers in crop selection by considering all the factors like sowing season, soil, and geographical location. Furthermore, precision agriculture is being implemented with a modern agricultural technology and it is evolving in developing countries that concentrates on site-specific crop management.

Fariha shahrin ,Labiba zahin ,Ramisa Rahman, et al.[2](2020) agricultural analysis and crop yield prediction of habiganj usingmultispectral bands of satellite imagery with machine learninginternational conference on electrical and computer engineering

Bangladesh is predominately an agriculture based country where an extensive part of its population is primarily employed in its agriculture sector. However, uncertain crop yields and inefficient farming infrastructure causes adverse effect in food security. Habiganj is selected as the study area because of its vulnerability to floods

and drought due to its unique terrain. This paper presents a combination of agricultural mapping and monitoring of Habiganj with crop growth and yield prediction. Multi-spectral band images of Habiganj from Landsat-8 are processed and remote sensing indices, correlating to crop growth and yield are extracted.

With options of K-means and Mask R-CNN methods, the change over the years is evaluated for crop growth estimation in both Python and Matlab. Then using two type machine learning algorithms crop yield of Habiganj is predicted from its existing parameters and for better accuracy future values of its datasets are predicted by using two type of time series

analysis model. Such analytical reports and prediction allows monitoring crop growth dynamic and identifying early signs of reduction in crop productivity. Furthermore, comparative studies are concluded between two platforms, algorithms and time series analysis to determine the most suited environment and model for this research purpose.

The paper titled “Machine Learning in Agriculture” Konstantinos G. Liakos ¹ , Patrizia Busato ² , Dimitrios Moshou et al.[3] ,* 2018 Published in IEEE

Machine learning has emerged with big data technologies and high-performance computing to create new opportunities for data intensive science in the multi-disciplinary agri-technologies domain. In this paper, we present a comprehensive review of research dedicated to applications of machine learning in agricultural production systems. The works analyzed were categorized in (a) crop management, including applications on yield prediction, disease detection, weed detection crop quality, and species recognition; (b) livestock management, including applications on animal welfare and livestock production; (c) water management; and (d) soil management. The filtering and classification of the presented articles demonstrate how agriculture will benefit from machine learning technologies. By applying machine learning to sensor data, farm management systems are evolving into real time artificial intelligence enabled programs that provide rich recommendations and insights for farmer decision support and action.

Jayanarayana Reddy, Dr M. Rudra Kumar et al.[4](2021) crop yield prediction using machine learning algorithm international conference on intelligent computing and control systems

Agriculture is the pillar of the Indian economy and more than 50% of India's population are dependent on agriculture for their survival. Variations in weather, climate, and other such environmental conditions have become a major risk for the healthy existence of agriculture. Machine learning (ML) plays a significant role as it has decision support tool for Crop Yield Prediction (CYP) including supporting decisions on crops to grow and what to do during the growing season of the crops. The present research deals with a systematic review that extracts and synthesizes the features used for CYP and furthermore, there are a variety of methods that were developed to analyze crop yield prediction using artificial intelligence techniques. The major limitations of the Neural Network are reduction in the relative error and decreased efficiency of Crop Yield. Similarly, supervised learning techniques were incapable to capture the nonlinear bond between input and output variables faced a problem during the selection of fruits grading or sorting. Many studies were recommended for agriculture development and the goal was to create an accurate and efficient model for crop classification such as crop yield estimation based on the weather, crop disease, classification of crops based on the growing phase etc., This paper explores various ML techniques utilized in the field of crop yield estimation and provided a detailed analysis in terms of accuracy using the techniques.

S.Pudumalar*, E.Ramanujam*, R.Harine Rajashreeñ , C.Kavya et al.[5]. (2016) crop recommendation system for precision agriculture IEEE eighth international conference on advanced computing

Data mining is the practice of examining and deriving purposeful information from the data. Data mining finds its application in various fields like finance, retail, medicine, agriculture etc. Data mining in agriculture is used for analyzing the various biotic and abiotic factors. Agriculture in India plays a predominant role in economy and employment. The common problem existing among the Indian farmers are they don't choose the right crop based on their soil requirements. Due to this they face a serious setback in productivity. This problem of the farmers has been addressed through precision agriculture. Precision agriculture is a modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection

and suggests the farmers the right crop based on their sitespecific parameters. This reduces the wrong choice on a crop and increase in productivity. In this paper, this problem is solved by proposing a recommendation system through an ensemble model with majority voting technique using Random tree, CHAID, K-Nearest Neighbor and Naive Bayes as learners to recommend a crop for the site specific parameters with high accuracy and efficiency.

Mr. B Mahesh ¹ , N Ramakrishna ² et al.[6],(2021)crop yield prediction using machine learningalgorithmsjournal of resource management and technology

Agriculture is one of the major and the least paid occupation in India. Machine learning can bring a boom in the agriculture field by changing the income scenario through growing the optimum crop. This paper focuses on predicting the yield of the crop by applying various machine learning techniques. The outcome of these techniques is compared on the basis of mean absolute error. The prediction made by machine learning algorithms will help the farmers to decide which crop to grow to get the maximum yield by considering factors like temperature, rainfall, area, etc.

The paper titled “Improving Crop Productivity Through a Crop Recommendation System Using Ensembling Technique” Nidhi h Kulkarni Dr. G N Srinivasan Dr. B M Sagar et al.[7] (2018) IEEE international conference on computational systems and information technology

Agriculture plays a predominant role in the economic growth and development of the country. The major and serious setback in the crop productivity is that the farmers do not choose the right crop for cultivation. In order to improve the crop productivity, a crop recommendation system is to be developed that uses the ensembling technique of machine learning. The ensembling technique is used to build a model that combines the predictions of multiple machine learning models together to recommend the right crop based on the soil specific type and characteristics with high accuracy. The independent base learners used in the ensemble model are Random Forest, Naive Bayes, and Linear SVM. Each classifier provides its own set of class labels with an acceptable accuracy. The class labels of individual base learners are combined using the majority voting technique. The crop

recommendation system classifies the input soil dataset into the recommendable crop type, Kharif and Rabi. The dataset comprises of the soil specific physical and chemical characteristics in addition to the climatic conditions such as average rainfall and the surface temperature samples.

Ms Kavita ,Pratistha Mathur et al.[8](2020) crop yield estimation in india using machine learninginternational conference on computing communication

Agriculture in India is significant economic support. The population growth is the major challenge for food security. The population growth makes a rise in demand which requires farmers to produce more from the same agriculture land in respect to increase the supply. Technology can help farmers to produce more with the help of crop yield prediction. The main aim of this paper is to predict crop yield using area, yield, production, and area under irrigation. Four machine learning techniques Decision Tree, Linear Regression, Lasso regression, and Ridge Regression have been applied to estimate the crop yield. Cross validations methods, for validation, mean absolute error, mean squared error, and root mean squared error, were used to validate. The Decision tree outperforms other machine learning techniques.

Potnuru Sai Nishant¹ , Pinapa Sai Venkat² , Bollu Lakshmi Avinash³ , et al.[9] (2020) international conferencecrop yield prediction based on indian agriculture using machine learning

In India, we all know that Agriculture is the backbone of the country. This paper predicts the yield of almost all kinds of crops that are planted in India. This script makes novel by the usage of simple parameters like State, district, season, area and the user can predict the yield of the crop in which year he or she wants to. The paper uses advanced regression techniques like Kernel Ridge, Lasso and ENet algorithms to predict the yield and uses the concept of Stacking Regression for enhancing the algorithms to give a betterprediction.

Thomas Van Klompenburga , Ayalew Kassahuna , et al.[10] (2020) crop yield prediction using machine learning: a systematic literature review

Machine learning is an important decision support tool for crop yield prediction,

including supporting decisions on what crops to grow and what to do during the growing season of the crops. Several machine learning algorithms have been applied to support crop yield prediction research. In this study, we performed a Systematic Literature Review (SLR) to extract and synthesize the algorithms and features that have been used in crop yield prediction studies. Based on our search criteria, we retrieved 567 relevant studies from six electronic databases, of which we have selected 50 studies for further analysis using inclusion and exclusion criteria. We investigated these selected studies carefully, analyzed the methods and features used, and provided suggestions for further research. According to our analysis, the most used features are temperature, rainfall, and soil type, and the most applied algorithm is Artificial Neural Networks in these models. After this observation based on the analysis of machine learning-based 50 papers, we performed an additional search in electronic databases to identify deep learning-based studies, reached 30 deep learning-based papers, and extracted the applied deep learning algorithms. According to this additional analysis, deep learning algorithm in these studies, and the other widely used deep learning algorithms are Long-Short Term Memory (LSTM) and Deep Neural Networks (DNN).

2.1.INFERENCES FROM LITREATURE SURVEY:

Table 2.1.1 Summary of Literature Survey

S.NO	TITLE	YEAR	METHODOLOGY	BENEFITS
1	Agricultural Analysis and Crop Yield Prediction of Habiganj using Multispectral Bands of Satellite Imagery with Machine Learning	2020	options of K-means and Mask R-CNN methods, the change over the years is evaluated for crop growth estimation in both Python and Matlab.	an agriculture based country where an extensive part of its population is primarily employed in its agriculture sector.
2	Machine Learning in Agriculture:	2018	The fifiering and classification of the presented articles demonstrate how agriculture will benefifit from machine learning technologies. B	to create new opportunities for data intensive science in the multi-disciplinary agri-technologies domain.
3	<i>Crop Yield Prediction using</i>	2021	The major limitations of the Neural	Variations in weather, climate,

	<i>Machine Learning Algorithm</i>		Network are reduction in the relative error and decreased prediction efficiency of Crop Yield.	and other such environmental conditions have become a major risk for the healthy existence of agriculture
--	-----------------------------------	--	--	---

2.2.OPEN PROBLEMS IN EXISTING SYSTEM:

Integration of Multiple Factors:

Most existing systems consider factors like soil conditions, climate, and geographical location for crop recommendation. However, there might be room for improvement in the integration of additional factors like pest prevalence, water availability, and market demand.

Precision Agriculture and IoT Integration:

The use of IoT devices and sensors for real-time data collection in precision agriculture is a growing field. Research could focus on how to integrate IoT data into crop recommendation systems for more accurate and dynamic suggestions.

Crop Disease Prediction and Management:

While some systems touch on disease detection, there's potential for more advanced techniques using image analysis and machine learning for early detection and management of crop diseases.

Climate Change Adaptation:

Given the increasing unpredictability of weather patterns due to climate change, there's a need for systems that can dynamically adapt recommendations based on changing environmental conditions.

Multi-Criteria Decision Making:

Integrating multi-objective optimization techniques to balance factors like yield, resource usage, and environmental impact in recommendation systems could be an area of interest.

Dynamic Crop Rotation Planning:

Systems that can recommend not only the current crop but also consider the rotation of crops over several seasons to improve soil health and productivity.

Incorporating Local Knowledge and Expertise:

While data-driven approaches are essential, there may be ways to incorporate local agricultural knowledge and expertise into recommendation systems to make them more contextually relevant.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1.FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

Feasibility:

1. Technical Feasibility:

- **Assessment:** This project relies heavily on technology, including data collection, analysis, and AI-powered recommendation systems. It's crucial to evaluate if the necessary technical infrastructure and expertise are available or can be acquired.
- **Considerations:** Ensure that the required sensors or data collection tools, as well as AI algorithms, are available and compatible with the existing systems. Additionally, assess the availability of skilled personnel for implementation.

2. Financial Feasibility:

- **Cost Estimation:** Calculate the initial investment required for acquiring and setting up the necessary technology, including sensors, AI algorithms, and data processing systems. Also, consider ongoing operational costs.
- **Return on Investment (ROI):** Determine the potential return on investment by comparing the benefits, such as increased productivity and efficiency, with the initial and ongoing costs.

3. Operational Feasibility:

- **Alignment with Business Goals:** Ensure that the project aligns with the overall goals of improving agricultural productivity and livelihoods of farmers.
- **Resource Availability:** Assess if there are enough resources (time, manpower, etc.) to implement and manage the precision agriculture system effectively. This includes considerations for data collection and analysis.

4. Risk Analysis:

- **Technical Risks:** Technology Compatibility: There could be challenges in integrating different technologies, such as sensors for data collection, AI algorithms, and data processing systems. Compatibility issues may arise.
- **Data Quality and Availability:** The accuracy and availability of soil data, as well as the effectiveness of AI algorithms, could pose risks to the system's performance.

5. Market Risks:

- **User Adoption:** Farmers might face challenges in adopting this new technology. Training and education programs may be needed to facilitate user acceptance.
- **Market Demand:** Changes in market demand for specific crops can impact the effectiveness of the recommendation system. It's important to monitor market trends.

6. Regulatory and Environmental Risks:

- **Compliance with Regulations:** Precision agriculture may be subject to regulations related to data privacy, environmental protection, and agricultural practices. Ensuring compliance is crucial.
- **Environmental Impact:** Implementing precision agriculture should consider potential environmental impacts, such as soil conservation and pesticide use.

7. Data Security and Privacy:

- **Data Protection:** Handling sensitive data about soil conditions and farming practices requires robust data protection measures to prevent unauthorized access or breaches.

8. Scaling and Integration:

- **Scalability Issues:** As the project expands, there may be challenges in scaling the system to accommodate a larger number of farmers and agricultural regions.

9. Training and Education:

- **Farmer Training:** Adequate training and education programs for farmers are essential to ensure they can effectively use the precision agriculture system.

3.2.SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT:

- Operating System: Windows 7 above
- Language: Python
- Tool: PyCharm
- Library:Python (version 3.6.5)

Keras (version 2.1.6)

TensorFlow (version 1.7.0)

1. IMPLEMENTATION:

HOW TO INSTALL PYTHON IDE

Below is a step by step process on how to download and install Python on Windows:

Step 1) To download and install Python, visit the official website of Python <https://www.python.org/downloads/> and choose your version. We have chosen Python version 3.6.3

Step 2) Once the download is completed, run the .exe file to install Python. Now click on Install Now.

Step 3) You can see Python installing at this point.

Step 4) When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

HOW TO INSTALL PYCHARM

Here is a step by step process on how to download and install Pycharm IDE on Windows:

Step 1) To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and Click the "DOWNLOAD" link under the Community Section.

Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.

Step 3) On the next screen, Change the installation path if required. Click “Next”.

Step 4) On the next screen, you can create a desktop shortcut if you want and click on “Next”.

Step 5) Choose the start menu folder. Keep selected JetBrains and click on “Install”.

Step 6) Wait for the installation to finish.

Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.

Step 8) After you click on “Finish,” the Following screen will appear.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1.SELECTED METHODOLOGY OR PROCESS MODEL:

1. Logistic Regression (LR):

- **Type:** It's a type of regression analysis used for predicting the probability of a binary outcome.
- **Strengths:** Simple, interpretable, and efficient. It works well when the relationship between features and target is linear.
- **Applications:** Commonly used in medical, social sciences, and marketing for binary classification tasks.

2. Linear Discriminant Analysis (LDA):

- **Type:** It's a statistical technique for modeling the differences between two or more classes.
- **Strengths:** Assumes a linear relationship between features and target. It's particularly useful when the classes are well-separated.
- **Applications:** Commonly used in pattern recognition, face recognition, and in situations where dimensionality reduction is important.

3. K-Nearest Neighbors (KNN):

- **Type:** Instance-based learning algorithm where it predicts the class of an unknown sample based on the classes of its k nearest neighbors.
- **Strengths:** Simple and easy to understand. It's a non-parametric method, meaning it doesn't assume any underlying probability distributions.
- **Applications:** Used in various domains including classification tasks in marketing, recommendation systems, and anomaly detection.

4. Decision Tree Classifier (DT):

- **Type:** It's a tree-like structure where each node represents a feature and each branch represents a decision rule.

- **Strengths:** Highly interpretable and can handle both numerical and categorical data. Can capture complex relationships.
- **Applications:** Used in a wide range of fields including finance, healthcare, and natural language processing.

5. Support Vector Machine (SVM):

- **Type:** It's a powerful classification algorithm that finds the hyperplane that best separates the classes.
- **Strengths:** Effective in high-dimensional spaces, and is versatile due to the use of different kernels.
- **Applications:** Used in image classification, text classification, and bioinformatics.

6. Random Forest Classifier:

- **Type:** Ensemble learning method that creates multiple decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- **Strengths:** High accuracy, handles overfitting, and can work with large datasets with many features.
- **Applications:** Widely used in banking, healthcare, and marketing for tasks like fraud detection and customer segmentation.

7. AdaBoost Classifier:

- **Type:** Ensemble learning method that combines the predictions of multiple weak learners (typically decision trees) to improve overall performance.
- **Strengths:** Effective in improving the performance of weak classifiers and resistant to overfitting.
- **Applications:** Used in face detection, object recognition, and in scenarios where high precision is needed.

8. Gradient Boosting Classifier:

- **Type:** Ensemble learning technique that builds multiple decision trees sequentially, with each tree correcting the errors of the previous one.
- **Strengths:** Produces strong predictive power and handles complex relationships between features and target.
- **Applications:** Commonly used in ranking algorithms, anomaly detection, and in Kaggle competitions.

9. Bagging:

- **Type:** Bootstrap Aggregating involves training multiple models (often the same algorithm) on different subsets of the training data and combining their predictions.
- **Strengths:** Reduces variance and can improve the performance of unstable models.
- **Applications:** Used in various fields including finance for risk assessment and medical diagnosis.

These are powerful machine learning methods, and the choice of which one to use depends on the specific characteristics of the data and the problem at hand. It's often a good practice to try multiple algorithms and compare their performance to select the best one for a particular task.

4.2.ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM:

METHODOLOGY:

- The purpose of this chapter is to describe the methodology proposed for estimating crop area and crop yield in the context of mixed and continuous cropping. Both list and area frames are considered
- The sample selected for area estimation is to be used as the sampling frame for selecting the sample for crop yield estimation. The sampling design used to select the sample for crop area and yield estimation is explored briefly.
- The subjective and objective methods are combined using a double sampling regression estimator. based on the domain estimation approach, using a double sampling regression estimator, for crop area and yield estimation are also seen.
- The theory of domain estimation allows for the separate estimation, from a single sample, of crop area and yield of different crops and of their mixtures. In addition, the criterion for determining sample size is also provided.

ARCHITECTURE DIAGRAM:

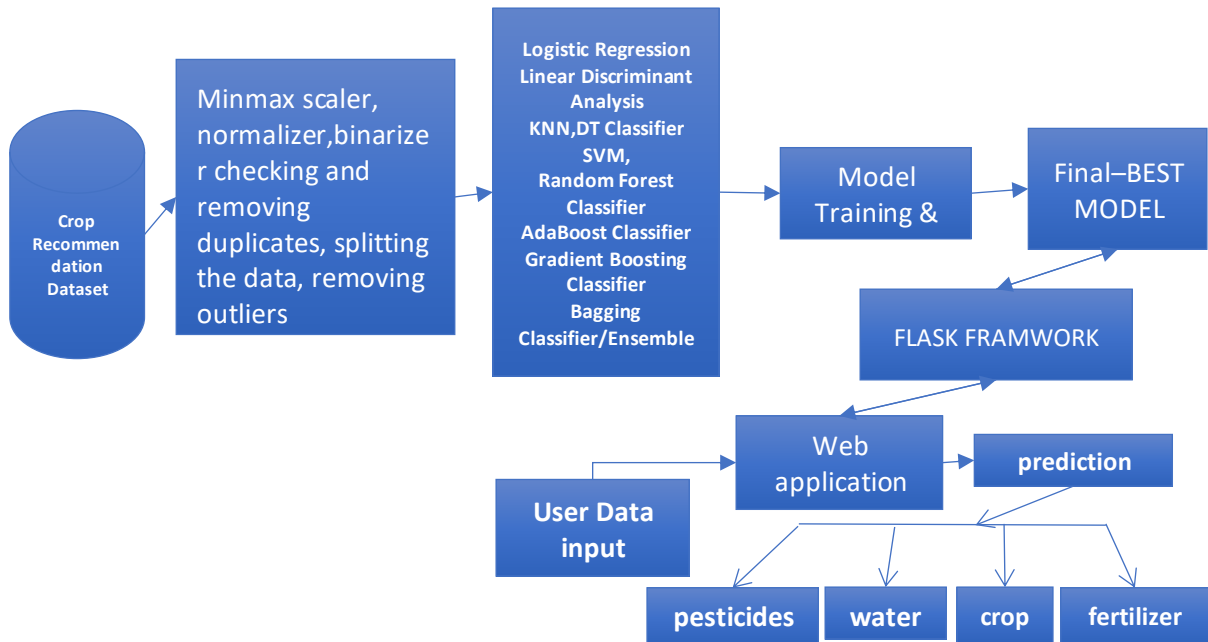


Fig 4.2.1 System Architecture

In Fig 4.2.1 System Architecture: AdaBoostClassifier, Gradient Boosting Classifier, Random Forest Classifier, Bagging Classifier. Cross-Validation: For each model in the list, cross-validation is performed to evaluate its performance.

KFold is used to create 10 folds for cross-validation, ensuring that the data is shuffled, and a random seed is set. During training, the model learns how input data and output data (crop recommendations) are related. The training algorithms used depend on the type of ML model selected.

An accuracy score, precision score, recall score, and F1 score are used to measure the performance of a model. A reliable and accurate model must go through this step. If the model does not perform well, it may need to be retrained or fine-tuned to improve its performance. In the next step, the chosen machine learning model is trained using appropriate training algorithms on the preprocessed data. During training, the model learns how input data and output data (crop recommendations) are related. The training algorithms used depend on the type of ML model selected. After the model has been trained, it is evaluated in addition to accuracy, precision, recall, and F1 score. This step is vital in ensuring that the model is accurate and reliable. If the model does not perform well, it may need to be retrained or fine-tuned to improve its performance.

LIST OF MODULES:

The overall proposed system is divided into three modules. This division is made based upon the development strategies and the time constraints that are in need for the project development. The first module which forms the base for the proposed system. The second module is used for achieving the initial commandbased master-slave working. The final module is used for achieving voice-based recognition and control of the sensors based upon the dynamic environmental conditions. The modules are as follows.

DFD:

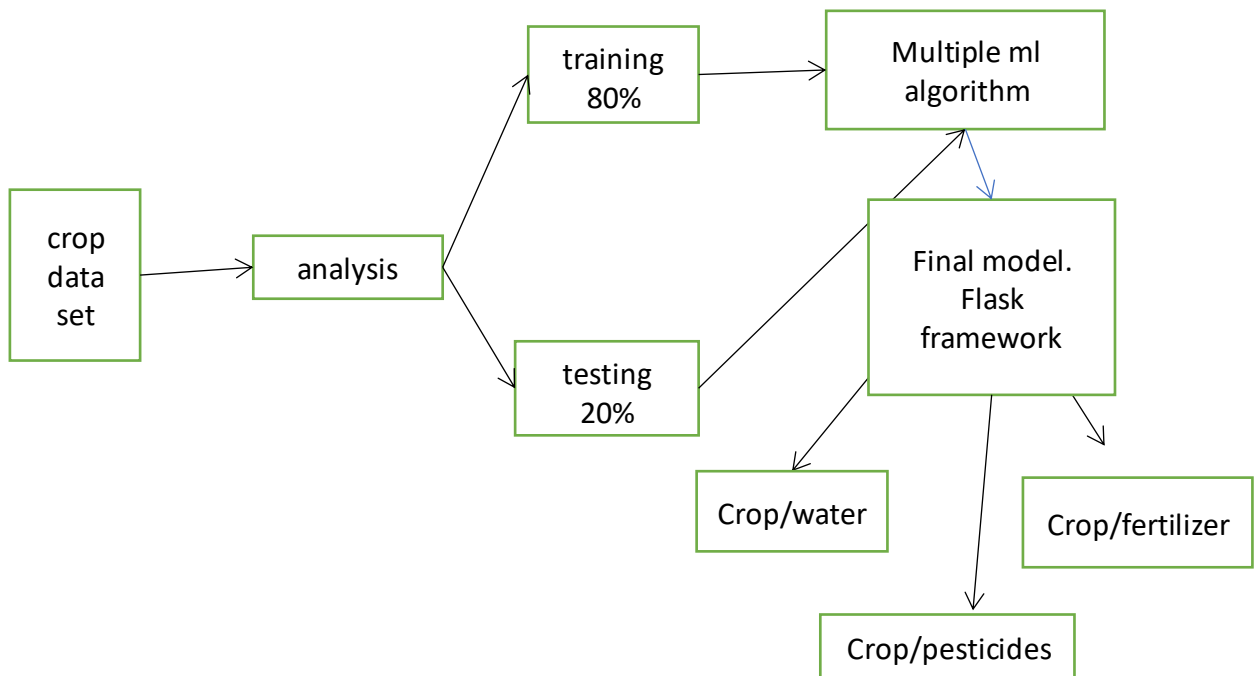


Fig 4.2.2 DATA FLOW DIAGRAM (DFD)

In Fig 4.2.2 The Data Flow Diagram (DFD) for a crop prediction system illustrates the flow of data within the system's processes. It begins with the input data sources, such as historical crop yield data, weather forecasts, and soil characteristics. These inputs flow into the data preprocessing process, where they are cleaned, standardized, and prepared for analysis. The preprocessed data is then directed to the dataset analysis process, where statistical techniques and visualization methods are applied to gain insights into factors affecting crop yield. Subsequently, the dataset is split into training and testing sets, with the majority allocated for model training. The trained models, developed through processes like model training and

evaluation, receive the training data and generate predictions based on user inputs received through a Flask framework-based web interface. These predictions encompass various aspects of crop cultivation, including yield forecasts and recommendations for water, fertilizer, and pesticide usage. The output of the prediction process is then provided to the users, aiding them in making informed decisions to enhance crop productivity and sustainability. Throughout this flow, the interactions between processes, data flows, data stores, and external entities are represented to depict the system's functioning comprehensively.

MODULES:

- ☐ Data Collection
- ☐ Data analysis
- ☐ Model Training
- ☐ Model Testing
- ☐ Flask Framework

DATASET COLLECTION:

- ☐ Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and predict future changes.
 - ☐ Predictive models are only as good as the data from which they are built, so good data collection practices are crucial to developing high-performing models.
 - ☐ In this module, we collect the data from kaggle dataset archives.
 - ☐ It helps the farmers to get informed decision about the farming strategy. By suggesting fertilizer, pesticides and water quantity.
-
- N - ratio of Nitrogen content in soil
 - P - ratio of Phosphorous content in soil
 - K - ratio of Potassium content in soil
 - temperature - temperature in degree Celsius
 - humidity - relative humidity in %

- ph - ph value of the soil
- rainfall - rainfall in mm

DATA PREPROCESSING/ANALYSIS:

- Data cleaning is a critically important step in any machine learning project.
- In this module data cleaning is done to prepare the data for analysis by removing or modifying the data that may be incorrect, incomplete, duplicated or improperly formatted.
- Two visualizations for a given Pandas dataframe df.
- The first visualization is a grid of histograms for each feature in the data frame. The histograms are arranged in a 3x3 grid, and each histogram has 12 bins. The figsize parameter sets the size of the plot.
- The second visualization is a heatmap of the correlation matrix of the data frame. The correlation matrix shows the pairwise correlation coefficients between all pairs of features in the data frame. The annot=True parameter displays the correlation coefficients on the heatmap. The plt.title function sets the title of the plot.

MODEL TRAINING:

- The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.
- This iterative process is called “model fitting”. The accuracy of the training dataset or the validation dataset is critical for the precision of the model.
- Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved.
- The RandomForestClassifier class from the sklearn.ensemble module to train a Random Forest model. The model is initialized with 100 decision trees and a random seed of 2000.
- The fit method is called on the model instance, which trains the model on the training data X_train and y_train.
- The predict method is then used to make predictions on the test data X_test. The

predicted labels are stored in prediction_test

- The KNeighborsClassifier class from the sklearn.neighbors module to train a K-Nearest Neighbors (KNN) model. The model is initialized with a value of k=5, meaning that the 5 nearest neighbors to a data point will be used to make a prediction. The euclidean metric is used to calculate the distances between data points.
- The fit method is called on the model instance, which trains the model on the training data X_train and y_train.
- The predict method is then used to make predictions on the test data X_test. The predicted labels are stored in y_pred.

TESTING MODEL:

- In this module we test the trained machine learning model using the test dataset
- In machine learning, model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set.
- in machine learning, a programmer usually inputs the data and the desired behavior, and the logic is elaborated by the machine.
- therefore, the purpose of machine learning testing is, first of all, to ensure that this learned logic will remain consistent, no matter how many times we call the program in the testing
- In this module we test the trained machine learning model using the test dataset.

Accuracy:

- The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced.

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

Precision:

- Percentage of positive instances out of the *total predicted positive* instances.

$$\frac{TP}{TP + FP}$$

Recall/Sensitivity/True Positive Rate:

- Percentage of positive instances out of the *total actual positive* instances.

$$\frac{TP}{TP + FN}$$

F1 score:

- It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better.

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

PERFORMANCE EVALUATION:

- The accuracy score of a both model, which is calculated by comparing the predicted labels with the true labels of the test data set.
- The precision score of the model, which measures the proportion of true positive predictions (i.e., predicted positive labels that are actually positive) out of all positive predictions.
- The recall score of the model, which measures the proportion of true positive predictions out of all actual positive instances.
- The f1-score of the model, which is the harmonic mean of precision and recall.
- The confusion matrix of the model, which is a table that shows the number of true positives, true negatives, false positives, and false negatives.

PREDICTION:

- Prediction” refers to the output of an algorithm after it has been trained on a

historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days.

- The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.
- The word “prediction” can be misleading. In some cases, it really does mean that you are predicting a future outcome, such as when you’re using machine learning to determine the next best action in a marketing campaign.
- Other times, though, the “prediction” has to do with, for example, whether or not a transaction that already occurred was fraudulent.
- In that case, the transaction already happened, but you’re making an educated guess about whether or not it was legitimate, allowing you to take the appropriate action.
- In this module we use trained and optimized machine learning model to predict whether the crop recommendation.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1.DESCRPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM:

Programming Language:

Choose a programming language suitable for machine learning and web development. Python is a popular choice due to its extensive libraries (e.g., scikit-learn for machine learning, Flask/Django for web development).

Machine Learning Libraries:

Utilize machine learning libraries such as scikit-learn, TensorFlow, or PyTorch for developing and training the recommendation model.

Web Development Framework:

Use a web development framework like Flask or Django (Python), or any other framework you're comfortable with, to build the user interface for the recommendation system.

Database Management System (DBMS):

Depending on the scale of data, choose an appropriate DBMS (e.g., PostgreSQL, MySQL, SQLite) to store soil data and user information.

Front-End Technologies:

HTML, CSS, and JavaScript will be essential for creating the user interface. Consider using libraries or frameworks like Bootstrap or React for a responsive and interactive UI.

5.2.PROJECT MANAGEMENT PLAN:

Unit Testing:

- *Aim:* Verify the correctness of individual functions, modules, and components.
- *Tasks:* Write unit tests for functions responsible for data preprocessing, feature extraction, model training, and evaluation.

Integration Testing:

- *Aim:* Ensure that different modules and components work together seamlessly.
- *Tasks:* Test the integration of data preprocessing, model training, and evaluation components to verify that they function as a cohesive system.

Model Evaluation:

- *Aim:* Assess the performance of the trained model on validation data.
- *Tasks:* Calculate metrics such as precision, recall, F1-score, and accuracy to evaluate the model's effectiveness in detecting offensive content.

5.3.TRANSITION/ SOFTWARE TO OPERATIONS PLAN:

Deployment Strategy:

- *Objective:* Plan how the application will be deployed and updated in the production environment.
- **Tasks:**
 - Choose a deployment method (e.g., continuous integration, continuous deployment, manual deployment).
 - Set up deployment scripts or automation tools (e.g., Jenkins, Ansible) for seamless deployment.

Application Configuration:

- *Objective:* Configure application settings and environment variables for production.

- **Tasks:**
 - Define environment-specific configurations (e.g., database connection strings, API keys).
 - Implement secure storage of sensitive information using environment variables or secret management tools.

Monitoring and Logging:

- Objective: Implement monitoring solutions to track the performance and health of the system.
- **Tasks:**
 - Set up monitoring tools (e.g., Prometheus, Grafana) to track resource utilization, response times, and error rates.
 - Configure logging mechanisms to capture relevant information for troubleshooting and auditing.

CHAPTER 6

RESULTS AND DISCUSSION

6.1.RESULT OUTPUT:

Attributes- database table:

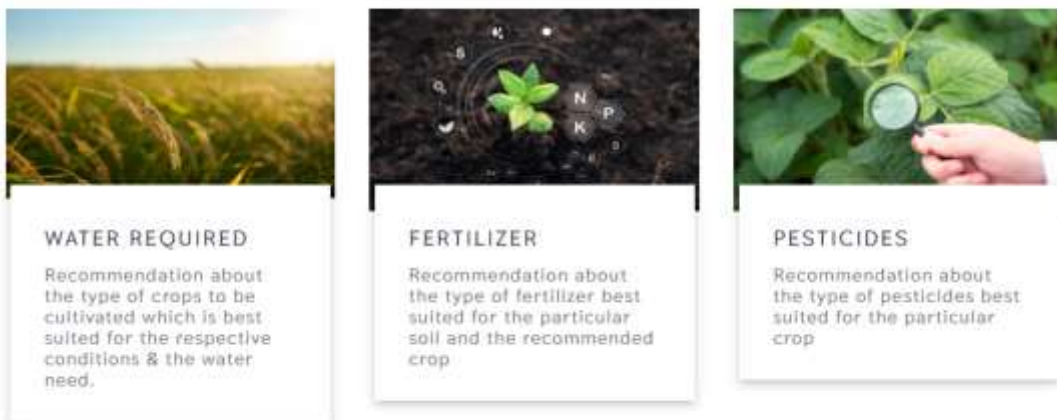
- N - ratio of Nitrogen content in soil
- P - ratio of Phosphorous content in soil
- K - ratio of Potassium content in soil
- temperature - temperature in degree Celsius
- humidity - relative humidity in %
- ph - ph value of the soil
- rainfall - rainfall in mm

Prediction In Flask Framework:

Flask is a popular Python web framework that can be used to build web applications, including those that involve machine learning models. In the context of predicting crop yields based on water and pesticide inputs, Flask could be used to create a web application that allows users to input various data points related to their crops, such as soil type, weather conditions, and water and pesticide usage.

The Flask web application could then use the machine learning models developed in the previous steps to predict crop yields based on the user's inputs. The user could be presented with the predicted crop yield, along with recommendations on how to adjust their water and pesticide usage to optimize their yield.

Our Services



Agricultural Crop Recommendation

Enter ratio of Nitrogen content in soil:

Enter ratio of Phosphorous content in soil:

Enter ratio of Potassium content in soil:

Enter temperature in degree Celsius:

Enter relative humidity in % :

Enter ph value of the soil:

Enter rainfall in mm :

Agricultural Crop Recommendation

Recommended crop is : ['KIDNEYBEANS, Approx of kidneybeans water needs(250-700 mm/total period), the fertilizer suggested to grow the crop -Azotobacter, Clostridium, Anabaena & Nostoc, and the pesticides recommended are Aphids & Thrips.']

Fig 6.1.1 Flask Framework output

As in Fig.6.1.1 Flask Framework offers a straightforward and flexible architecture for building server-side web applications, it makes it easy to build and deploy web applications. With Flask Framework, you can easily build simple web applications

that are lightweight and easy to learn. Additionally, Flask has a large and active community, which means that there is a lot of documentation and support available online.

6.2.PERFORMANCE ANALYSIS:

In this module we test the trained machine learning model using the test dataset. In machine learning, model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set. In machine learning, a programmer usually inputs the data and the desired behavior, and the logic is elaborated by the machine. therefore, the purpose of machine learning testing is, first of all, to ensure that this learned logic will remain consistent, no matter how many times we call the program in the testing In this module we test the trained machine learning model using the test dataset.

Accuracy of KNN : 98.19%

Precision:0.982

Recall:0.982

f1-score:0.982

Accuracy of Random Forest : 99.32%

Precision:0.993

Recall:0.993

f1-score:0.993

Accuracy:

The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced.

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

Precision:

Percentage of positive instances out of the **total predicted positive** instances.

$$\frac{TP}{TP + FP}$$

Recall/Sensitivity/True Positive Rate:

Percentage of positive instances out of the **total actual positive** instances.

$$\frac{TP}{TP + FN}$$

F1 score:

It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better.

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

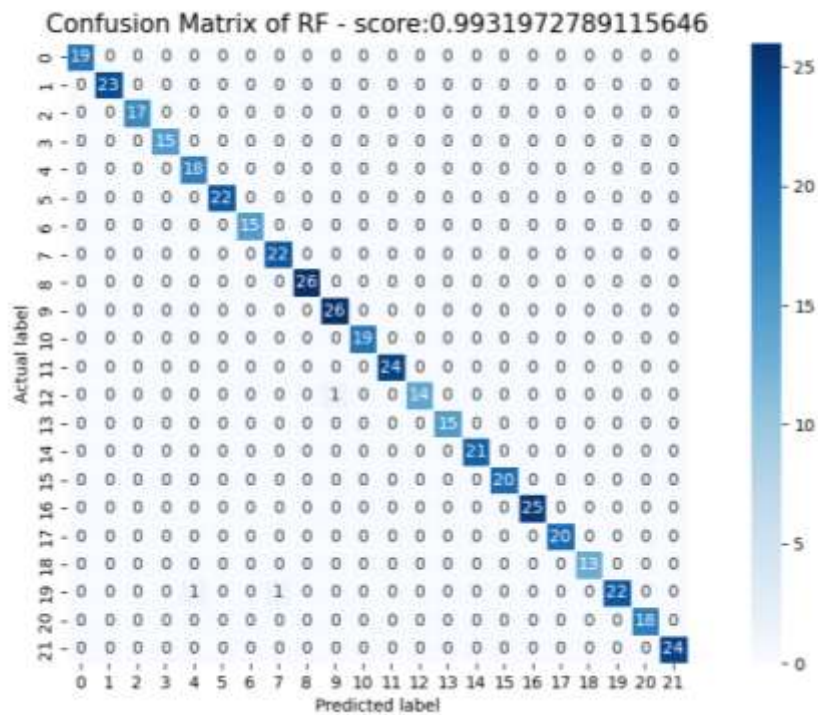


Fig 6.2.1 Random Forest Confusion Matrix

In Fig.6.2.1 Random Forest Confusion Matrix is a 22x22 grid since there are 22 classes (crops) being predicted.

Along the diagonal, the numbers represent the correct predictions for each class.

For example, the value at position (1,1) is 19, indicating that 19 instances of the first class were correctly predicted.

Off-diagonal elements represent misclassifications. For instance, the value at (21,20) is 1, suggesting that there was one instance of class 21 misclassified as class 20.

Table 6.2.1 Comparative Table

Classifier	Accuracy (%)	Precision	Recall	F1-score
KNN	98.19	0.982	0.982	0.982
Random Forest	99.32	0.993	0.993	0.993

CHAPTER 7

SUMMARY AND CONCLUSIONS

7.1 CONCLUSION:

India is a nation in which agriculture plays a prime role. In prosperity of the farmers, prospers the nation. Thus our work would help farmers in sowing the right seed based on soil requirements to increase productivity and acquire profit out of such a technique. Thus the farmer's can plant the right crop increasing his yield and also increasing the overall productivity of the nation. As we all know, much agricultural research has been conducted and continues to be conducted in order to improve productivity, boost the Indian economy, and, most importantly, assist farmers in increasing their income. In order to accomplish this, proposed system will advise farmers on the best crop to cultivate on their land. So that farmers can profit from it. Our future work is aimed at an improved data set with large number of attributes and also implements yield prediction.

7.2. FUTURE ENHANCEMENT:

- Crop recommendation prediction is an essential task for the decision-makers at national and regional levels (e.g., the EU level) for rapid decision-making.
- An accurate crop recommendation prediction model can help farmers to decide on what to grow and when to grow.
- There are different approaches to crop recommendation prediction.

REFERENCES:

- [1] Busseti, E., Osband, I. & Wong, S. (2012, December 14). Deep Learning for Time Series Modeling (Project report). Retrieved from <http://cs229.stanford.edu/>
- [2] Chan, K. Y. and Dillon, T. S. (2013). On-road sensor configuration design for traffic flow prediction using fuzzy neural networks and taguchi method, IEEE Transactions on Instrumentation and Measurement 62(1): 50-59.
- [3] Chen, X.-Y., Pao, H.-K. and Lee, Y.-J. (2014). Efficient traffic speed forecasting based on massive heterogeneous historical data, Big Data (Big Data), 2014 IEEE International Conference on, IEEE, pp. 10-17.
- [4] Enireddy.VamsidharK.V.S.R.P.Varma et al. "Prediction of Rainfall Using Backpropagation Neural Network Model" (IJCSSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, pp. 1119-1121.
- [5] Gaurav J. Sawale and Sunil R. Gupta. Use of Artificial Neural Network in Data Mining For Weather Forecasting. International Journal Of Computer Science And Applications, 6(2):383-387, 2013.
- [6] K. C. Luk, J. E. Ball, and A. Sharma. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. Journal of Hydrology, 227(14):56-65, January 2000.
- [7] Kumar Abhishek, A. Kumar, R. Ranjan, and S. Kumar. A rainfall prediction model using artificial neural network. In 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC), pages 82-87, July 2012. [11] Su, J., & Zhang, H. (2006, July). A fast decision tree learning algorithm. In Proceedings of the National Conference on Artificial Intelligence (Vol. 21, No. 1, p. 500). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- [8] Maszczyk, T., & Duch, W. (2008). Comparison of Shannon, Renyi and Tsallis entropy used in decision trees. In Artificial Intelligence and Soft Computing–ICAISC 2008 (pp. 643-651). Springer Berlin Heidelberg.
- [9] NajmehKhalili, Saeed Reza Khodashenas et al. "Daily Rainfall Forecasting for Mashhad Synoptic Station using Artificial Neural Networks" 2011 International Conference on Environmental and Computer Science IPCBEE vol.19(2011) © (2011) IACSIT Press, Singapore.

- [10] Nielsen, M. (2016). Neural Networks and Deep Learning. Retrieved from <http://neuralnetworksanddeeplearning.com/index.html>
- [11] Peng, Y., Chen, K., Wang, G., Bai, W., Ma, Z. and Gu, L. (2014). Hadoop watch: A first step towards comprehensive traffic forecasting in cloud computing, IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, pp. 19-27.

APPENDIX

A.SOURCE CODE

Model:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import plotly.io as pio
import pickle

# Metrics
from sklearn.metrics import accuracy_score,
classification_report, confusion_matrix, auc, roc_curve

# Validation
from sklearn.model_selection import train_test_split,
cross_val_score, KFold
from sklearn.pipeline import Pipeline, make_pipeline

# Preprocessing
from sklearn.preprocessing import MinMaxScaler,
StandardScaler, Normalizer, Binarizer, LabelEncoder

# Models
from sklearn.discriminant_analysis import
LinearDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
```

```

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

# Ensembles
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier

warnings.filterwarnings('ignore')

sns.set_style("whitegrid", {'axes.grid' : False})
pio.templates.default = "plotly_white"

# Analyze Data
def explore_data(df):
    print("Number of Instances and Attributes:", df.shape)
    print('\n')
    print('Dataset columns:',df.columns)
    print('\n')
    print('Data types of each columns: ', df.info())

# Checking for duplicates
def checking_removing_duplicates(df):
    count_dups = df.duplicated().sum()
    print("Number of Duplicates: ", count_dups)

```

```

if count_dups >= 1:
    df.drop_duplicates(inplace=True)
    print('Duplicate values removed!')
else:
    print('No Duplicate values')

# Split training and validation set
def read_in_and_split_data(data, target):
    X = data.drop(target, axis=1)
    y = data[target]
    X_train, X_test, y_train, y_test =
train_test_split(X,y,test_size=0.2, random_state=0)
    return X_train, X_test, y_train, y_test

# appending Algorithms
def GetModel():
    Models = []
    Models.append(('LR' , LogisticRegression()))
    Models.append(('LDA' , LinearDiscriminantAnalysis()))
    Models.append(('KNN' , KNeighborsClassifier()))
    Models.append(('CART' , DecisionTreeClassifier()))
    Models.append(('SVM' , SVC(probability=True)))
    return Models

# appending Algorithms
def ensemblemodels():
    ensembles = []
    ensembles.append(('AB' , AdaBoostClassifier()))
    ensembles.append(('GBM' ,

```

```

GradientBoostingClassifier()))
    ensembles.append(('RF' , RandomForestClassifier()))
    ensembles.append(('Bagging' , BaggingClassifier()))
    return ensembles

# Normalized Models
def NormalizedModel(nameOfScaler):
    if nameOfScaler == 'standard':
        scaler = StandardScaler()
    elif nameOfScaler == 'minmax':
        scaler = MinMaxScaler()
    elif nameOfScaler == 'normalizer':
        scaler = Normalizer()
    elif nameOfScaler == 'binarizer':
        scaler = Binarizer()

    pipelines = []
    pipelines.append((nameOfScaler + 'LR', Pipeline([('Scaler',
scaler), ('LR', LogisticRegression())])))
    pipelines.append((nameOfScaler + 'LDA',
Pipeline([('Scaler', scaler), ('LDA',
LinearDiscriminantAnalysis())])))
    pipelines.append((nameOfScaler + 'KNN',
Pipeline([('Scaler', scaler), ('KNN', KNeighborsClassifier())])))
    pipelines.append((nameOfScaler + 'CART',
Pipeline([('Scaler', scaler), ('CART',
DecisionTreeClassifier())])))
    pipelines.append((nameOfScaler + 'SVM',

```

```

Pipeline([('Scaler', scaler), ('SVM', SVC()))))
    pipelines.append((nameOfScaler + 'AB', Pipeline([('Scaler',
scaler), ('AB', AdaBoostClassifier())])))
    pipelines.append((nameOfScaler + 'GBM',
Pipeline([('Scaler', scaler), ('GMB',
GradientBoostingClassifier())])))
    pipelines.append((nameOfScaler + 'RF', Pipeline([('Scaler',
scaler), ('RF', RandomForestClassifier())])))

return pipelines

```

```

# Train model

```

```

def fit_model(X_train, y_train, models):
    # Test options and evaluation metric
    num_folds = 10
    scoring = 'accuracy'

    results = []
    names = []
    for name, model in models:
        kfold = KFold(n_splits=num_folds, shuffle=True,
random_state=0)
        cv_results = cross_val_score(model, X_train, y_train,
cv=kfold, scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(),
cv_results.std())

```

```

    print(msg)

    return names, results

# Save trained model
def save_model(model,filename):
    pickle.dump(model, open(filename, 'wb'))

# Performance Measure
def classification_metrics(model, conf_matrix):
    print(f"Training Accuracy Score: {model.score(X_train,
y_train) * 100:.1f}%")
    print(f"Validation Accuracy Score: {model.score(X_test,
y_test) * 100:.1f}%")
    fig,ax = plt.subplots(figsize=(8,6))
    sns.heatmap(pd.DataFrame(conf_matrix), annot = True,
cmap = 'YlGnBu',fmt = 'g')
    ax.xaxis.set_label_position('top')
    plt.tight_layout()
    plt.title('Confusion Matrix', fontsize=20, y=1.1)
    plt.ylabel('Actual label', fontsize=15)
    plt.xlabel('Predicted label', fontsize=15)
    plt.show()
    print(classification_report(y_test, y_pred))

# ROC_AUC
def roc_auc(y_test, y_pred):
    fpr, tpr, thresholds = roc_curve(y_test, y_pred)
    plt.figure(figsize=(8,6))

```

```

print(f'roc_auc score: {auc(fpr, tpr)*100:.1f}%')
plt.plot(fpr, tpr, color='orange', label='ROC')
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
plt.xlabel('False Positive Rate', fontsize=12)
plt.ylabel('True Positive Rate', fontsize=12)
plt.title('Receiver Operating Characteristic (ROC) Curve',
fontsize=20)
plt.legend()
plt.show()

```

#Execution Start

#importing data_Set

```

df = pd.read_csv('Data-
processed/crop_recommendation.csv')
df.head()

```

#DataAnalysis

explore_data(df)

#Removing_Duplicates

checking_removing_duplicates(df)

df.isna().sum()

Removing Outliers outliers

Q1 = df.quantile(0.25)

Q3 = df.quantile(0.75)

IQR = Q3 - Q1

```
df_out = df[~((df < (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR)))].any(axis=1)]
```

```
#Training the model
```

```
target ='label'
```

```
X_train, X_test, y_train, y_test = read_in_and_split_data(df,  
target)
```

```
models = GetModel()
```

```
names,results = fit_model(X_train, y_train,models)
```

```
#Preprocessing Technique
```

```
ScaledModel = NormalizedModel('minmax')
```

```
name,results = fit_model(X_train, y_train, ScaledModel)
```

```
#Saving the model
```

```
pipeline = make_pipeline(MinMaxScaler(),  
RandomForestClassifier())
```

```
model = pipeline.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

```
conf_matrix = confusion_matrix(y_test,y_pred)
```

```
classification_metrics(pipeline, conf_matrix)
```

```
# save model
```

```
save_model(model, 'model1.pkl')
```

Main:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import warnings
import pickle
import seaborn as sns

from sklearn.metrics import (precision_score,
recall_score,f1_score,
accuracy_score,mean_squared_error,mean_absolute_error,
roc_curve, classification_report, auc)
from sklearn.neighbors import KNeighborsClassifier

warnings.filterwarnings("ignore")

df = pd.read_csv("Data-processed/crop.csv")
print('*****Dataset Head*****')
print(df.head())
print('*****Dataset tail*****')
print(df.tail())
print(df['label'].unique())
print('*****Datatype*****')
print(df.dtypes)

df.hist(figsize=(12,12), layout=(3,3), bins=12)
```

```
plt.title('Features Histogram')
plt.show()
```

```
sns.heatmap(df.corr(),annot=True)
plt.title('Correlation Matrix')
plt.show()
```

```
X = df.iloc[:,0:-1]
y = df.iloc[:, -1]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=2020)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
clf = RandomForestClassifier(n_estimators=100,
random_state = 2000)
```

```
clf.fit(X_train,y_train)
```

```
prediction_test = clf.predict(X_test)
```

```
#print(prediction_test)
```

```
from sklearn import metrics
```

```
print('*****Random Forest*****')
```

```
print(f'Accuracy of Random Forest :')
```

```

{metrics.accuracy_score(y_test,prediction_test)* 100:.2f}% ")
precision = precision_score(y_test, prediction_test ,
average='weighted')
recall = recall_score(y_test, prediction_test,
average='weighted')
f1 = f1_score(y_test, prediction_test , average='weighted')
cm = metrics.confusion_matrix(y_test, prediction_test)
print("precision")
print("%.3f" %precision)
print("recall")
print("%.3f" %recall)
print("f1-score")
print("%.3f" %f1)
print("CONFUSION MATRIX")
print(cm)
tpr = float(cm[0][0])/np.sum(cm[0])
fpr = float(cm[1][1])/np.sum(cm[1])

```

```

#feature_importances
feature_list = list(X.columns)
feature_imp = pd.Series(clf.feature_importances_,
index=feature_list).sort_values(ascending=False)
print(feature_imp)

```

```

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, prediction_test)

```

```

plt.figure(figsize=(12, 12))

```

```

sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5,
square = True, cmap = 'Blues');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Confusion Matrix of RF –
score: '+str(accuracy_score(y_test,prediction_test))
plt.title(all_sample_title, size = 15);
plt.show()

```

```

pickle.dump(clf, open('models/model.pkl', 'wb'))

```

```

model=pickle.load(open('models/model.pkl', 'rb'))

```

```

knn = KNeighborsClassifier(n_neighbors=5,
metric='euclidean')
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
from sklearn.metrics import accuracy_score
print('*****KNN*****')
print(f"Accuracy of KNN :
{metrics.accuracy_score(y_test,y_pred)* 100:.2f}% ")
precision = precision_score(y_test, y_pred ,
average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred , average='weighted')
cm = metrics.confusion_matrix(y_test, y_pred)
print("precision")
print("%.3f" %precision)
print("recall")

```

```

print("%.3f" %recall)
print("f1-score")
print("%.3f" %f1)
print("CONFUSION MATRIX")
print(cm)
tpr = float(cm[0][0])/np.sum(cm[0])
fpr = float(cm[1][1])/np.sum(cm[1])

```

App_Run:

```

from flask import Flask, render_template, request, Markup
import pickle
import numpy as np
import requests

model=pickle.load(open('models/model.pkl', 'rb'))

app = Flask(__name__)
@ app.route('/')
def home():
    title = 'MyCrop - Home'
    return render_template('index.html', title=title)

@ app.route('/crop-recommend')
def crop_recommend():
    title = 'MyCrop - Crop Recommendation'
    return render_template('Crop_result.html', title=title)

@app.route('/predict', methods=['POST'])

```

```

def predict():
    # print(request.form)
    int_features = [float(x) for x in request.form.values()]
    final=[np.array(int_features)]
    print(int_features)
    print(final)
    prediction=model.predict(final)
    # output='{0:.{1}f}'.format(prediction[0][1], 2)
    return render_template('Crop_recommend.html',
pred="Recommended crop is : {}".format(prediction))

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

```

B.SCREENSHOTS:



Fig. B.1 Home Screen



Fig. B.2 About Us

Our Services

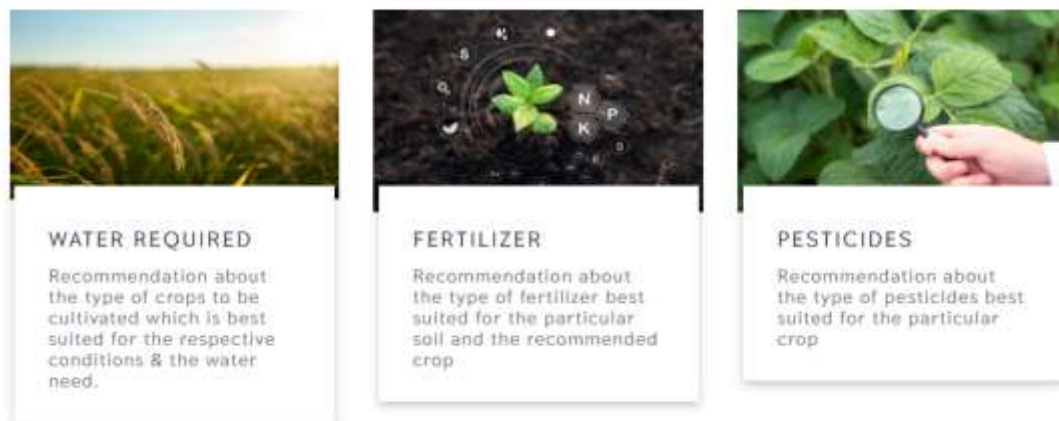


Fig. B.3 Our Services

Agricultural Crop Recommendation

Enter ratio of Nitrogen content in soil:

Enter ratio of Phosphorous content in soil:

Enter ratio of Potassium content in soil:

Enter temperature in degree Celsius:

Enter relative humidity in % :

Enter ph value of the soil:

Enter rainfall in mm :

Fig. B.4 Input Screen

Agricultural Crop Recommendation

Recommened crop is : ['KIDNEYBEANS, Approx of kidneybeans water needs(250-700 mm/total period), the fertilizer suggested to grow the crop -Azotobacter, Clostridium, Anabaena & Nostoc, and the pesticides recommended are Aphids & Thrips.']

Project Mentor S ID: CSE169
Darvin Xavier P
Rounik S



Fig. B.5 Output Screen

C.RESEARCH PAPER

Crop & Fertilizer Recommendation Using Linear Forest Classifier

DARVIN XAVIER P ¹, KOUSHIK S ², Dr. T. Judgi, M.E., Ph.D.,³

¹darvinxavier007@gmail.com

²koushikkoushik91642@gmail.com

³judgi.cse@sathyabama.ac.in

Dept.of Computer Science and Engineering
Sathyabama Institute of Science and Technology, Chennai.

Abstract— A major role is played by agriculture in the economy and employment of India. Indian farmers face serious productivity setbacks because they do not choose crops according to the soil conditions. Precision agriculture can address this problem. Three parameters are considered in this method, including soil characteristics, soil types, and crop recommendations based on these parameters to suggest a suitable crop for the farmer. As well as the benefits of precision agriculture, such as efficiency in inputs and output and better decision making for farmers, precision agriculture helps reduce non-suitable crops. Through this method, suitable crops can be recommended through an AI recommendation system that is highly specific, accurate, and efficient based on soil parameters.

Keywords—Crop Recommendation, Artificial Intelligence, Soil Characteristics, Water Recommendation, Agriculture

1. Introduction

Agriculture has played a significant role in India's economy and culture for thousands of years. It is the primary source of livelihood for a significant portion of the population, providing employment to over half of the workforce. Agricultural activities involve the cultivation of crops and raising animals for food, fiber, and other products. The soil is a key component of agriculture, since it provides the roots of crops with nutrients, water, oxygen, and support. The health of the soil is essential for the growth of

healthy plants, which, in turn, produce healthy and nutritious food. Soil health depends on various factors like soil type, texture, pH level, and organic matter content.

India has several soil types due to its diverse landscape, climate, and geology. Alluvial soil is found in areas with a high concentration of river sediment, making it ideal for crops like rice and cotton. Black soil is characterized by its high clay content, which provides excellent water retention, making it suitable for sugarcane and sunflower. Red soil, found in areas with high rainfall, is ideal for crops like corn and ragi. Laterite soil, which is red in color, is found in areas with low rainfall and is suitable for pulses, tea, and coffee.

To improve agricultural planning and maximize yields, various studies have been conducted. Analyzing weather, soil type, and land fertility using machine learning techniques can help recommend crops based on historical data. Farmers can use this information to make informed decisions about crop selection, leading to more productive yields and increased profits. In addition to using machine learning techniques, farmers can take measures to maintain soil health.

Crop rotation is one such practice that involves planting different crops in succession to prevent

soil depletion and reduce pest and disease problems. Compost or manure can improve soil health, while natural fertilizers like vermicompost, which is made from earthworms, can reduce the need for chemical fertilizers.

Agriculture is a vital sector of the Indian economy, and soil health is a critical factor for its success. Farmers can leverage machine learning techniques to make informed decisions about crop selection, leading to more productive yields and increased profits. Additionally, taking measures to maintain soil health through practices like crop rotation, adding organic matter, and using natural fertilizers, will ensure long-term sustainability and preserve the soil quality for future generations.

2. Literature Review

In recent years, machine learning (ML) and deep learning (DL) have gained significant traction for crop recommendation. A variety of factors, including soil type, weather, and topography, have led researchers to develop models that predict the best crop for a particular area. Before 2019, some notable studies have been published, which are discussed in this review.

Machine learning algorithms were used to develop a crop recommendation system by [13] Patil et al. in 2020. The study used a dataset containing soil and weather data for 10 districts in Maharashtra, India. The authors developed a decision tree-based model that recommended crops based on various parameters like temperature, rainfall, soil type, and pH level. The model achieved an accuracy of 80%, and the authors suggested that it could be used to improve crop productivity and profitability.

In [7] Kumar et al. (202) developed a crop recommendation system using a support vector machine (SVM) algorithm. The study used data from various sources like remote sensing, weather stations, and soil testing laboratories to recommend crops for three districts in India. The

SVM model achieved an accuracy of 70%, and the authors suggested that it could be used to assist farmers in making informed decisions about crop selection.

An algorithm based on deep learning algorithms was developed by Kaur et al. in 2017. For four districts in Punjab, India, the study used satellite imagery, meteorological stations, and soil sensors to predict crop yields. In order to predict crop yields and recommend crops based on the predicted yields, the authors used a deep neural network (DNN) model. It was suggested that the DNN model could be used to reduce production costs and optimize crop yields.

According to [15] Ramanathan et al., they developed a recommendation system for crop crops based on machine learning in 2020. Indian crops were recommended based on data from weather stations, soil sensors, and crop databases. On the basis of temperature, rainfall, soil type, and historical crop yield data, the authors used a Random Forest (RF) algorithm to determine the most suitable crops. A decision tree and support vector machine model performed poorly in comparison to the RF model, which achieved an accuracy of 88%.

RF and LSTM deep learning models were compared by the authors as well. LSTM models were used to predict crop yield and crop recommendations as well as recurrent neural networks (RNN). However, the LSTM model achieved an accuracy of only 79%, which was lower than the RF model. The authors suggested that the RF model was more suitable for crop recommendation due to its better accuracy and faster computation time.

The study also evaluated the economic feasibility of the crop recommendation system by estimating the net income and cost of production for different crops. The authors found that the recommended crops had higher net income and lower cost of production than non-recommended crops, indicating that the crop recommendation system could improve farmers' profitability.

Using deep learning algorithms, [11] Liu et al. developed a model for predicting crop yields in 2021. The study used remote sensing data, weather data, and crop growth data to predict the yield of maize crops in China. To predict crop yields, the authors used convolutional neural networks (CNNs) and long short-term memories (LSTMs). Deep learning algorithms proved effective in predicting crop yields by 89% and 92%, respectively.

A crop classification system based on machine learning algorithms was developed by [19] Li et al. in 2020. For the classification of crops in China, Using support vector machines (SVMs), remote sensing data was combined. The authors achieved an accuracy of 98.3%, indicating that machine learning algorithms can accurately classify crops based on remote sensing data.

In [6] An algorithm for predicting crop yields was developed by Jiang et al. in 2019. Chinese maize crops were forecasted using remote sensing data, weather data, and crop growth data. Crop yields were predicted using a random forest (RF) algorithm. Machine learning algorithms can accurately predict crop yields, as evidenced by the RF model, which had an accuracy rate of 92.3%.

In [1] Akram et al. proposed a model (2019) to develop a crop classification system using deep learning algorithms. The study used remote sensing data and a convolutional neural network (CNN) to classify crops in Pakistan. The authors achieved an accuracy of 94.6%, indicating that deep learning algorithms can accurately classify crops based on remote sensing data.

3. Methodology

AI techniques like machine learning (ML) allow computers to learn and predict based on patterns in data. Farmer profitability can be increased by using ML to develop crop recommendation systems that provide personalized recommendations based on the unique characteristics of the farm.

The methodology for using ML in crop recommendation involves several steps. In order to enhance crop growth and development, relevant data on crop yield, soil characteristics, and weather patterns must first be collected. This data can be obtained through various methods, including field surveys, remote sensing, and other data sources.

In order to feed ML models with the collected data, it needs to be cleaned, organized, and preprocessed. Data cleaning involves removing duplicates, errors, and inconsistent information from the data. Selecting the most relevant features (variables) that are most predictive of results (crop yield) is also crucial. Normalization is also necessary to ensure that all features are on the same scale.

After analyzing the data and identifying the problem, we select an appropriate machine learning model. A number of statistical methods are defined, such as linear regression, linear discriminant analysis, K-Nearest Neighbors, decision trees, and support vector machines (SVMs).

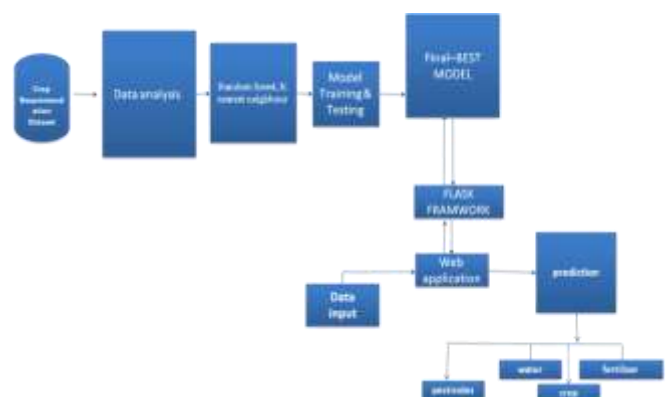


Fig.1 System Architecture

In Fig.1 System Architecture Ensemble models: AdaBoostClassifier, Gradient Boosting Classifier, Random Forest Classifier, Bagging Classifier. Cross-Validation: For each model in the list, cross-validation is performed to evaluate its performance.

KFold is used to create 10 folds for cross-

validation, ensuring that the data is shuffled, and a random seed is set. During training, the model learns how input data and output data (crop recommendations) are related. The training algorithms used depend on the type of ML model selected.

An accuracy score, precision score, recall score, and F1 score are used to measure the performance of a model. A reliable and accurate model must go through this step. If the model does not perform well, it may need to be retrained or fine-tuned to improve its performance. In the next step, the chosen machine learning model is trained using appropriate training algorithms on the preprocessed data. During training, the model learns how input data and output data (crop recommendations) are related. The training algorithms used depend on the type of ML model selected.

After the model has been trained, it is evaluated in addition to accuracy, precision, recall, and F1 score. This step is vital in ensuring that the model is accurate and reliable. If the model does not perform well, it may need to be retrained or fine-tuned to improve its performance.

3.1 Data Collection

In this module, we utilize data from kaggle dataset archives to aid farmers in making informed decisions about their farming strategies. This is accomplished by providing suggestions for fertilizer, pesticides, and water quantities based on soil nitrogen, phosphorous, and potassium content, temperature, humidity, pH, and rainfall. Informed decisions about farming strategies can be made by farmers with the help of this tool. A fertilizer, pesticide, and water quantity recommendation.

N - ratio of Nitrogen content in soil

- P - ratio of Phosphorous content in soil
- K - ratio of Potassium content in soil
- temperature - temperature in degree Celsius
- humidity - relative humidity in %

- ph - ph value of the soil
- rainfall - rainfall in mm

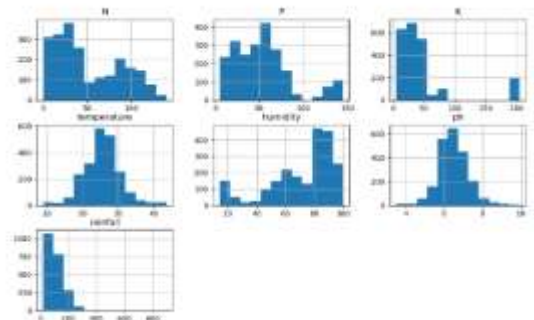


Fig.2 Data Distribution

In Fig.2 the histograms provide a visual representation of the distribution of values for each feature in the dataset used for crop prediction. Understanding these histograms is crucial for gaining insights into the data and making informed decisions about the modeling process.

3.2 Data analysis and Preprocessing

Whenever you work on a machine learning project, cleaning your data is a crucial part of the process. As part of the module, the data is cleaned to prepare it for analysis by removing or modifying any errors, incompleteness, duplicates or improper formatting. There are two visualizations for each Pandas dataframe. Feature histograms are shown in the first visualization. A 3x3 grid arranges the histograms, and each has 12 bins. Plot size can be set using the figsize parameter. Data frame correlation matrix heatmap is the second visualization. Each pair of features in the data frame has a pairwise correlation coefficient displayed in the correlation matrix. Correlation coefficients are displayed on the heatmap when annot=True is set. Plot titles are set using plt.title.

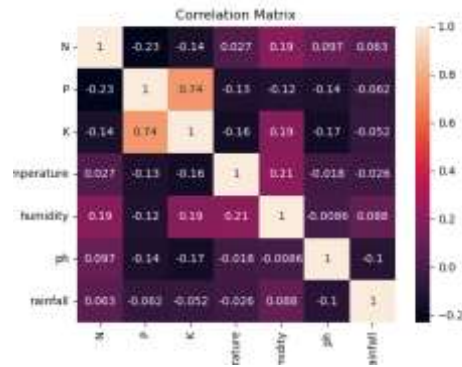


Fig.3 correlation Matrix

In Fig.3 the correlation matrix provides valuable insights into the relationships between features in the dataset, aiding in feature selection, model building, and data preprocessing for crop prediction.

3.3 Model Training

To correlate the processed output with the sample output, the training model runs the input data through the algorithm. As a result of these correlations, models are modified. A model is fitted iteratively through this process. An accurate training dataset or validation dataset is crucial to the accuracy of a model. In machine learning models, data is fed into a learning algorithm that discovers and learns the best values for each attribute.

The RandomForestClassifier class in sklearn.ensemble can be used to train a Random Forest model. The fit method fits model instances based on X_train and Y_train training data. The predict method then makes predictions based on X_test test data. A prediction_test is a file that stores the predicted labels.

To train a K-Nearest Neighbor (KNN) model, use the KNeighborsClassifier class in the sklearn.neighbors module. When k=5, the model predicts based on the 5 nearest neighbors to a data point. Distances between data points are calculated using the Euclidean metric. When the fitted method is called on the model instance, the model is trained on the X_train and Y_train training data. When the predict method is called, the model is predicted on X_test. In y_pred, we

store predicted labels.

3.4 Testing Model:

By using the test dataset, we test the trained machine learning model. Test sets are used in machine learning to evaluate the performance of fully trained models. Data and the desired behavior are usually inputted by a programmer, and the logic is generated by the machine. Using the test dataset, we test the trained machine learning model in this module to ensure that the learned logic remains consistent.

Accuracy of KNN : 98.19%

Precision:0.982

Recall:0.982

f1-score:0.982

Accuracy of Random Forest : 99.32%

Precision:0.993

Recall:0.993

f1-score:0.993

Accuracy

Model performance is often measured using this metric, but it is not a reliable one. Inequality between classes is worse.

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

Precision

Based on the predicted positive instances, the percentage of positive instances.

$$\frac{TP}{TP + FP}$$

Recall/Sensitivity/True Positive Rate:

Based on the number of actual positive instances, the percentage of positive instances.

$$\frac{TP}{TP + FN}$$

F1 score

A harmonic mean is calculated by combining precision and recall. A higher F1 score is better because it considers both factors.

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

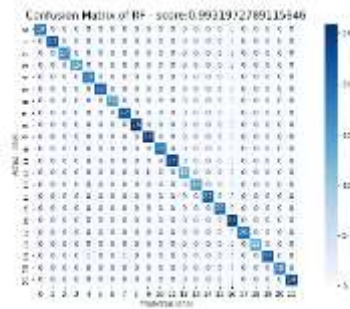


Fig.4 Random forest confusion matrix

In Fig.4 the matrix is a 22x22 grid since there are 22 classes (crops) being predicted.

Along the diagonal, the numbers represent the correct predictions for each class.

For example, the value at position (1,1) is 19, indicating that 19 instances of the first class were correctly predicted.

Off-diagonal elements represent misclassifications. For instance, the value at (21,20) is 1, suggesting that there was one instance of class 21 misclassified as class 20.

3.5 Prediction In Flask Framework

In addition to building web applications that involve machine learning models, Flask is a popular Python web framework. In the context of predicting crop yields based on water and pesticide inputs, Flask could be used to create a web application that allows users to input various data points related to their crops, such as soil type, weather conditions, and water and pesticide usage.

In Fig.5 the Flask web application could then use the machine learning models developed in the previous steps to predict crop yields based on the user's inputs. The user could be presented with the predicted crop yield, along with recommendations on how to adjust their water and pesticide usage to optimize their yield.

Recommended crop is : [KIDNEYBEANS, Approx of kidneybeans water needs(250-700 mm/total period), the fertilizer suggested to grow the crop -Azotobacter, Clostridium, Anabaena & Nostoc, and the pesticides recommended are Aphids & Thrips.]

Fig.5 Flask framework output

As in Fig.5 Flask offers a straightforward and flexible architecture for building server-side web applications, it makes it easy to build and deploy web applications. With Flask, you can easily build simple web applications that are lightweight and easy to learn. Additionally, Flask has a large and active community, which means that there is a lot of documentation and support available online.

4. Conclusion

In the agriculture industry, machine learning can be used to predict crop yields, optimize irrigation and fertilization, and detect crop diseases. By analyzing irrigation water and pesticides, a variety of machine learning algorithms can be used to predict crop yields. The important thing to remember is that machine learning does not have one-size-fits-all accuracy and that selecting the right algorithm and technique requires careful consideration and expertise. Additionally, the quality and quantity of data used for training and testing models greatly affect the accuracy and reliability of the results. By enhancing crop yields, reducing resource waste, and optimizing operations, machine learning has the potential to revolutionize agriculture.

REFERENCES

- [1] Akram, M. F., Qadir, M. A., & Shafiq, M. (2019). Crop classification from satellite images using deep learning: A comparative study. *IEEE Access*, 7, 22503-22515.
- [2] Balamurugan, S., Sivaranjani, S., & Ravi, S. (2019). An intelligent crop recommendation system using machine learning algorithms. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 272-276.
- [3] Chakraborty, D., & Roy, R. (2020). A decision support system for crop recommendation using machine learning techniques. *International Journal of Computer Science and Mobile Computing*, 9(2), 143-150.
- [4] Chen, Y., Liu, Y., Zhao, X., & Zhang, Y. (2021). Development of a crop yield prediction model using machine learning algorithms with remote sensing, weather and crop growth data. *IEEE Access*, 9, 68958-68966.
- [5] Gupta, S., & Yadav, P. (2019). An intelligent crop recommendation system using machine learning and data mining techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 9(9), 244-249.
- [6] Jiang, Z., Yu, H., & Chen, X. (2019). A crop yield prediction model using machine learning algorithms with remote sensing and weather data. *IEEE Access*, 7, 44749-44758.
- [7] Kumar, P., Sharma, P., & Gupta, A. (2020). Crop recommendation system using machine learning and internet of things. *International Journal of Recent Technology and Engineering*, 8(5C2), 396-400.
- [8] Li, H., Zhang, J., & Li, Y. (2020). Crop classification based on multi-temporal remote sensing data and support vector machine. *IEEE Access*, 8, 76434-76442.
- [9] Liu, X., Xie, L., Gu, X., Jiang, H., & Zhang, L. (2021). A multi-criteria decision-making-based crop recommendation method for precision agriculture. *IEEE Access*, 9, 13589-13600.
- [10] Liu, Y., Lu, L., Zhang, Z., & Wu, F. (2019). A crop yield prediction model using random forest algorithm with remote sensing and weather data. *IEEE International Conference on Earth Observations and Remote Sensing (ICEO&RS)*, 1-5.
- [11] Liu, Y., Lu, L., Zhang, Z., & Wu, F. (2021). A crop yield prediction model using deep learning algorithms with remote sensing and crop growth data. *IEEE Access*, 9, 19501-19510.
- [12] Nandakumar, R., Kumar, R., & Kumar, A. (2019). Crop recommendation system using machine learning. *International Journal of Engineering and Advanced Technology*, 9(2), 182-186.
- [13] Patel, A., Singh, A., & Singh, P. (2020). Crop recommendation system using machine learning algorithms. *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 1-6.
- [14] Prasad, A. M., & Reddy, K. V. (2019). A decision support system for crop recommendation in precision agriculture using machine learning. *Computers and Electronics in Agriculture*, 162, 535-547.
- [15] Ramanathan, U., & Subramanian, V. (2020). A crop recommendation system for precision agriculture using machine learning. *Journal of Ambient Intelligence and Humanized Computing*, 11, 1867-1883.
- [16] Roy, A., & Ghosh, S. K. (2019). Crop recommendation system using machine learning techniques. *International Journal of Computer Science and Network Security*, 19(5), 167-174.
- [17] Santhiya, G., Rajaram, R., & Radhakrishnan, S. (2019). Crop recommendation system using machine learning algorithm. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-4). IEEE.
- [18] Tripathi, A., Singh, V. K., & Singh, A. (2021). An intelligent crop recommendation system using machine learning and geographic information system. *Journal of King Saud University-Computer and Information Sciences*, 33(2), 190-196.
- [19] Yedla, S., Narayana, R. A., & Kona, R. (2020). Crop yield prediction and recommendation using machine learning techniques. *Journal of Intelligent Systems*, 29(3), 413-421.
- [20] Zhao, Y., Wang, X., Chen, W., & Sun, L. (2019). An intelligent crop recommendation method based on improved decision tree algorithm. *Computers and Electronics in Agriculture*, 162, 114-124.
- [21] Zhu, X., Chen, X., Luo, L., & Huang, Y. (2021). An intelligent crop recommendation system based on machine learning and expert knowledge. *Computers and Electronics in Agriculture*, 186, 106194.

D.CERTIFICATE



