PREDICTION OF CROP YIELD USING MACHINE LEARNING ALGORITHM

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

By

KATTA PRANAY (Reg.No - 39110480)
KOVVURI SAI SRIKANTH REDDY (Reg.No - 39110536)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | 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 **KATTA PRANAY(Reg.No - 39110480) and KOVVURI SAI SRIKANTH REDDY(Reg.No - 39110536)** who carried out the Project Phase-2 entitled **"PREDICTION OF CROP YIELD USING MACHINE LEARNING ALGORITHM"** under my supervision from January 2023 to April 2023.

yovanfelix

Internal Guide

Dr. A. YOVAN FELIX M.E., Ph.D

Head of the Department

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

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY DEPARTMENT OF CSE

Submitted for Viva voce Examination held on 2.5.2023

InternalExaminer **ExternalExaminer**

DECLARATION

I, **KATTA PRANAY (Reg.No- 39110480),** hereby declare that the Project Phase-2 Report entitled**"PREDICTION OF CROP YIELD USING MACHINE LEARNING ALGORITHM"** done by me under the guidance of **Dr.A.YOVAN FELIX,M.E.,Ph.D** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

KATTA PRANAY

DATE: 02-05-2023

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** 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.A.YOVAN FELIX .M.E.,Ph.D,**for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks toall 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

Agriculture is the field that assumes a significant part in improving our nation's economy. Farming is the one that brought forth human advancement. India is an agrarian country and its economy generally dependent on crop productivity. Agriculture is the spine of all business in our country. Choosing a crop is vital in Agriculture arranging. The determination of crops will rely on the various boundaries, for example, market value, production rate and distinctive government policies. Numerous progressions are needed in the agriculture field to improve changes in our Indian economy. Improvements in agriculture can be done utilizing machine learning techniques which are applied effectively on cultivating area. Alongside all advances in the machines and innovations utilized in cultivating, valuable and exact data about various matters likewise assumes a huge part in it. The aim of the proposed system is to carry out the yield determination technique using Decision Tree Regressor, Random forest with the goal that this strategy helps in taking care of numerous agriculture and farmers issues. This improves our Indian economy by expanding the yield rate of crop production.

TABLE OF CONTENTS

`

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

Abbrevation	Full Form
ICAR	Indian Council of Agricultural Research
SVM	Support vector machine
NB	Naïve Baye’s
KNN	K-Nearest Neighour
CNN	Convolutional Neural Networks
CSM	Crop Selection Method
RELU	Rectified Linear activation unit

CHAPTER 1

INTRODUCTION

In the world of developing technologies, the success of sharing information will help the agriculturists in realizing and developing their potential. Crop yield estimation is an important aspect of agriculture, as it helps farmers and policymakers make informed decisions related to crop management, food security, and trade. In recent years, there has been an increasing interest in the use of machine learning techniques to improve crop yield estimation in India.Machine learning algorithms are able to process large amounts of data, and can be trained to recognize patterns in data that may be difficult for humans to detect. This makes them well-suited for analyzing the vast amounts of data that are generated in agriculture, such as climate data, soil data, and satellite imagery.

One approach to using machine learning for crop yield estimation is to develop predictive models that can estimate crop yields based on a range of inputs, such as weather data, soil moisture data, and historical crop yield data. These models can be trained using supervised learning techniques, where the model is trained on historical data to predict future outcomes.Another approach is to use unsupervised learning techniques, such as clustering and anomaly detection, to identify patterns in large datasets. For example, satellite imagery can be analyzed using clustering techniques to identify regions with similar crop growth patterns, which can be used to estimate crop yields.There have been several initiatives in India to develop machine learning-based crop yield estimation systems. For example, the Indian Council of Agricultural Research (ICAR) has developed a crop yield estimation system that uses machine learning algorithms to predict crop yields based on weather and soil data.

Overall, the use of machine learning techniques for crop yield estimation in India has the potential to significantly improve agriculturalproductivity and food security, by providing accurate and timely information to farmers and policymakers.The information sharing is that the valuable and timely information is being shared between agriculturists, either formally or informally. The willingness of information sharing refers to the open attitude among agriculturists. This open attitude determines the degree and scope of information sharing. Using web-technologies like html and css we build the web application, We create dataset by gathering data from multiple resources and place them in place which is used to predict the price of the crop and results are subjected to non-linear test later priorities are set and rankings are given to the list of crops. Place information in our application and share that information to agriculturists whose data is collected and stored in the mysql server. we software to automatically send the updated information to the agriculturists in the form of text message.so that agriculturists no need to go to near by towns and cities to know the updated information. We will be machine learning algorithms to predict the price of the crop for the next two months. For prediction purpose we will be using Support vector machine(SVM), Naïve Baye's (NB) and K-Nearest Neighour(KNN) algorithms to predict the cost of the crop production. Further, a ranking process is applied for decision making in order to select the classifiers results.

Agriculture is the backbone of the Indian economy, providing employment to over 50% of the population and contributing approximately 16% of the country's GDP. Crop yield estimation plays a vital role in agriculture, as it enables farmers and policymakers to make informed decisions related to crop management, food security, and trade. In recent years, there has been a growing interest in the use of machine learning techniques to improve crop yield estimation in India.This paper aims to provide an overview of crop yield estimation in India, the challenges faced in estimating crop yields, and the potential of machine learning techniques to address these challenges. The paper will also discuss some of the initiatives undertaken in India to develop machine learning-based crop yield estimation systems.

Challenges in Crop Yield Estimation in India Estimating crop yields in India is a challenging task, primarily due to the following reasons:

Heterogeneous agro-climatic conditions: India has diverse agro-climatic conditions, with variations in rainfall, temperature, soil type, and crop varieties across different regions. These variations make it difficult to develop a generalized model for crop yield estimation that can be applied across the country.

Limited availability of data: The availability of reliable data is a significant challenge in estimating crop yields. The data related to weather, soil, and crop growth is often

limited, inconsistent, and fragmented. This makes it difficult to develop accurate crop
Human errors: Crop yield estimation in India is primarily done through manual surveys, which are prone to errors due to human biases and limitations. These errors can result in inaccurate estimations, which can have significant implications for farmers and policymakers.

Lack of timely information: Timely information is crucial for effective crop management, but the current crop yield estimation methods in India often provide delayed information, making it difficult for farmers to take appropriate measures to manage their crops.Machine Learning Techniques for Crop Yield Estimation in India
Machine learning techniques have the potential to overcome some of the challenges faced in crop yield estimation in India. Machine learning algorithms can process large amounts of data, and can be trained to recognize patterns in data that may be difficult for humans to detect. This makes them well-suited for analyzing the vast amounts of data that are generated in agriculture, such as climate data, soil data, and satellite imagery.
There are several approaches to using machine learning for crop yield estimation in India:
Predictive modeling: One approach is to develop predictive models that can estimate crop yields based on a range of inputs, such as weather data, soil moisture data, and historical crop yield data. These models can be trained using supervised learning techniques, where the model is trained on historical data to predict future outcomes. For example, a model can be developed to predict wheat yields based on temperature, rainfall, and soil moisture data. Such models can be further improved by incorporating additional data sources, such as satellite imagery and drone data.

Unsupervised learning: Another approach is to use unsupervised learning techniques, such as clustering and anomaly detection, to identify patterns in large datasets. For example, satellite imagery can be analyzed using clustering techniques to identify regions with similar crop growth patterns, which can be used to estimate crop yields. Anomaly detection can be used to identify regions with abnormal crop growth patterns, which can be indicative of crop stress or disease outbreaks.

Deep learning: Deep learning techniques, such as convolutional neural networks (CNNs), can be used to analyze satellite imagery and extract features related to crop growth. These features can be used to estimate crop yields or identify areas with abnormal crop growth patterns. CNNs have been used to analyze satellite imagery to estimate wheat yields in India, achieving higher accuracy than traditional crop yield estimation methods.

In conclusion, the use of machine learning techniques for crop yield estimation in India has the potential to revolutionize agriculture in the country. These techniques can overcome the challenges associated with traditional crop yield estimation methods and provide accurate and timely information to farmers and policymakers. While there are still some challenges to be overcome, such as the availability and quality of data, the potential benefits of machine learning-based crop yield estimation systems are significant and worth pursuing.

CHAPTER 2

LITERATURE SURVEY

[1]Title : Rice crop yield prediction using Support Vector Machines-2019

Authors : Sunil Kumar, Vivek Kumar, R. K. Sharma

In the domain of Soft Computing, Support Vector Machines (SVMs) have acquired considerable significance. These are widely used in making predictions, owing to their ability of generalization. This paper is about the development of SVM based classification models for the prediction of rice yield in India. Experiments have been conducted involving oneagainst-one multi classification method, k-fold cross validation and polynomial kernel function for SVM training. Rice production data of India has been sourced from Directorate of Economics and Statistics, Ministry of Agriculture, Government of India, for this work. The best prediction accuracy for the 4- year relative average increase has been achieved as 75.06% using 4-fold cross validation method. MATLAB software has been used for experimentation in this work.

[2]Title : Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique-2015

Authors : Rakesh Kumar, M. P. Singh, Prabhat Kumar, J. P. Singh

Agriculture planning plays a significant role in economic growth and food security of agro-based country. Se- lection of crop(s) is an important issue for agriculture planning. It depends on various parameters such as production rate, market price and government policies. Many researchers studied prediction of yield rate of crop, prediction of weather, soil classification and crop classification for agriculture planning using statistics methods or machine learning techniques. If there is more than one option to plant a crop at a time using limited land resource, then selection of crop is a puzzle. This paper proposed a method named Crop Selection Method (CSM) to solve crop selection problem, and maximize net yield rate of crop over season and subsequently achieves maximum economic growth of the country. The proposed method may improve net yield rate of crops.

[3]Title : Crop Prediction system using machine Learning- 2020

Authors : Nischitha K, Dhanush Vishwakarma, Mahendra N, Ashwini, Manjuraju

As we know the fact that, India is the second largest population country in the world and majority of people in India have agriculture as their occupation. Farmers are growing same crops repeatedly without trying new verity of crops and they are applying fertilizers in random quantity without knowing the deficient content and quantity. So, this is directly affecting on crop yield and also causes the soil acidification and damages the top layer. So, we have designed the system using machine learning algorithms for betterment of farmers. Our system will suggest the best suitable crop for particular land based on content and weather parameters. And also, the system provides information about the required content and quantity of fertilizers, required seeds for cultivation. Hence by utilizing our system farmers can cultivate a new variety of crop, may increase in profit margin and can avoid soil pollution.

[4]Title : Crop yield Prediction using Machine Learning Techniques-2019

Authors : Aruvansh Nigam; Saksham Garg; Archit Agrawal; Parul Agrawal

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.

[5]Title : A Machine Learning Approach to Predict Crop Yield and Success Rate-2019

Author: Shivani S. Kale; Preeti S. Patil

In India agriculture contributes approximately 23% of GDP and employed workforce percentage is 59%. India is the second-largest producer of agriculture crops. the technological contribution may help the farmer to get more yield. The prediction of the yield of different crops may help the farmer regarding taking the decision about which crop to grow. The research focuses on the prediction of different crops yield using neural network regression modeling. The data of crop cycle for summer, Kharif, rabi, autumn and whole year is used. The dataset is resourced from an Indian government website. The experimental parameters considered for study are cultivation area, crop, state, district, season, year and production or yield for the period of 1998 to 2014. The dataset consists of 2 lakh 40 thousand records. The dataset is filtered using Python Pandas and Pandas Profiling tools to retrieve data for Maharashtra state. The model is developed using a Multilayer perceptron neural network. Initially the result obtained considering optimizer RMS prop with accuracy 45 %, later it will be enhanced to 90% by increasing layers, adjusting weight, bias and changing optimizer to Adam. This research describes the development of a different crop yield prediction model with ANN, with 3 Layer Neural Network. The ANN model develops a formula to ascertain the relationship using a large number of input and output examples, to establish model for yield predictions an Activation function: Rectified Linear activation unit (Relu) is used. The backward and forward propagation techniques are used.

[6]Title: Applications of Machine Learning Techniques in Agricultural Crop Production-2016

Author: Subhadra Mishra, Debahuti Mishra, and Gour Hari Santra

This paper has been prepared as an effort to reassess the research studies on the relevance of machine learning techniques in the domain of agricultural crop production. **Methods/Statistical Analysis:** This method is a new approach for production of agricultural crop management. Accurate and timely forecasts of crop production are necessary for important policy decisions like import-export, pricing marketing distribution etc. which are issued by the directorate of economics and statistics. However one has understand that these prior estimates are not the objective estimates as these estimate requires lots of descriptive assessment based on many different qualitative factors. Hence there is a requirement to develop statistically sound objective prediction of crop production. That development in computing and information storage has provided large amount of data.

Findings: The problem has been to intricate knowledge from this raw data , this has lead to the development of new approach and techniques such as machine learning that can be used to unite the knowledge of the data with crop yield evaluation. This research has been intended to evaluate these innovative techniques such that significant relationship can be found by their applications to the various variables present in the data base.

Application / Improvement: The few techniques like artificial neural networks, Information Fuzzy Network, Decision Tree, Regression Analysis, Bayesian belief network. Time series analysis, Markov chain model, k-means clustering, k nearest neighbor, and support vector machine are applied in the domain of agriculture were presented.

[7]Title : Machine Learning: Applications in Indian Agriculture-2016

Authors : Karandeep Kaur

Agriculture is the mainstay of a developing economy like India. Majority of its population depends on agriculture for their income. With depleting resources, reducing land sizes and increase in input and labor costs, combined with the uncertainty of various factors like weather, market prices etc, agriculture in India has become a profession which is full of risks. The advancements in technology must be worked upon across various disciplines and it has already shown dramatic improvements in many fields. However, agriculture has not benefitted much from such advancements. Smart farming is the need of the hour of the Indian economy. Machine learning is an imminent field of computer science which can be applied to the farming sector quite effectively. It can facilitate the up-gradation of conventional farming techniques in the most cost-friendly approach. The purpose of this paper is to broaden the farming horizon by listing and evaluating the different applications of machine learning in Indian agriculture and to help the farmers advance their work up by many notches.

CHAPTER3
Requirement Analysis

3.1 Feasibility Studies/Risk Analysis of the Project

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.1.2 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.1.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.1.4 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Project crop yield prediction using machine learning algorithms like decision tree algorithm and random forest algorithm has the potential to be socially feasible. Here are some reasons why:

- Farmers can benefit from this technology by using it to make better decisions about when to plant, how much to plant, and when to harvest their crops. This can lead to increased crop yields, which can help to reduce food insecurity and improve the overall economic well-being of farmers.
- Governments and other organizations can also benefit from this technology by using it to identify areas of the country where crop yields are likely to be low. This can help them to target resources and support to these areas, which can help to reduce poverty and improve overall agricultural productivity.
- By using machine learning algorithms like decision tree and random forest, this technology can be made more accurate and reliable over time. This can help to build trust in the technology among farmers and other stakeholders, which can lead to greater adoption and use.
- As this technology becomes more widespread, it can help to promote greater transparency and accountability in the agricultural sector. By providing farmers with more accurate and timely information about crop yields, it can help to reduce fraud and corruption in the sector.

Overall, project crop yield prediction using machine learning algorithms like decision tree algorithm

3.2 MODULES

- Data Collection
- Calculate yield of production
- Predict crop value
- Accuracy on test set

Data Collection

• This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

• There are several techniques to collect the data, like web scraping, manual interventions and etc.

• The dataset used in this crop yield prediction in India taken from some other source

Calculate yield of production

• In this project, crops price is calculated by quality of the crop is identified using ranking process. By this process the min and max rate of crop production is also notified.

Predict crop value

• In this module the crop value is predicted by applying machine learning algorithms to the collected and train data. So that we can know the crop min max value of the crop at any particular area i.e based on the input.

Accuracy on test set

• We got a accuracy of 90.7% on test set.

DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.1
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

LEVEL-0

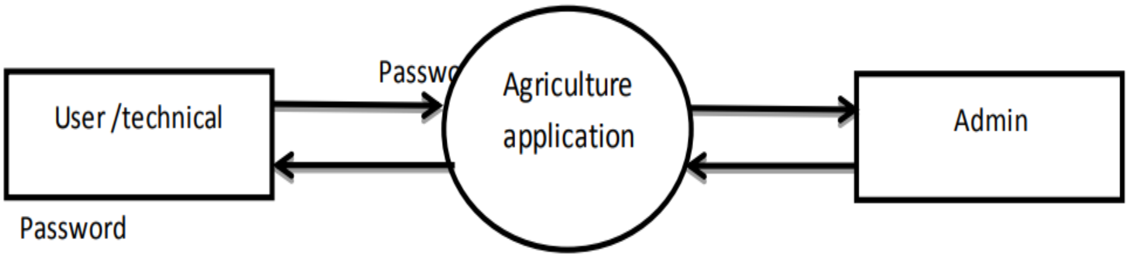


Fig. 3.1 Data Flow Diagram Level-0

Level-1

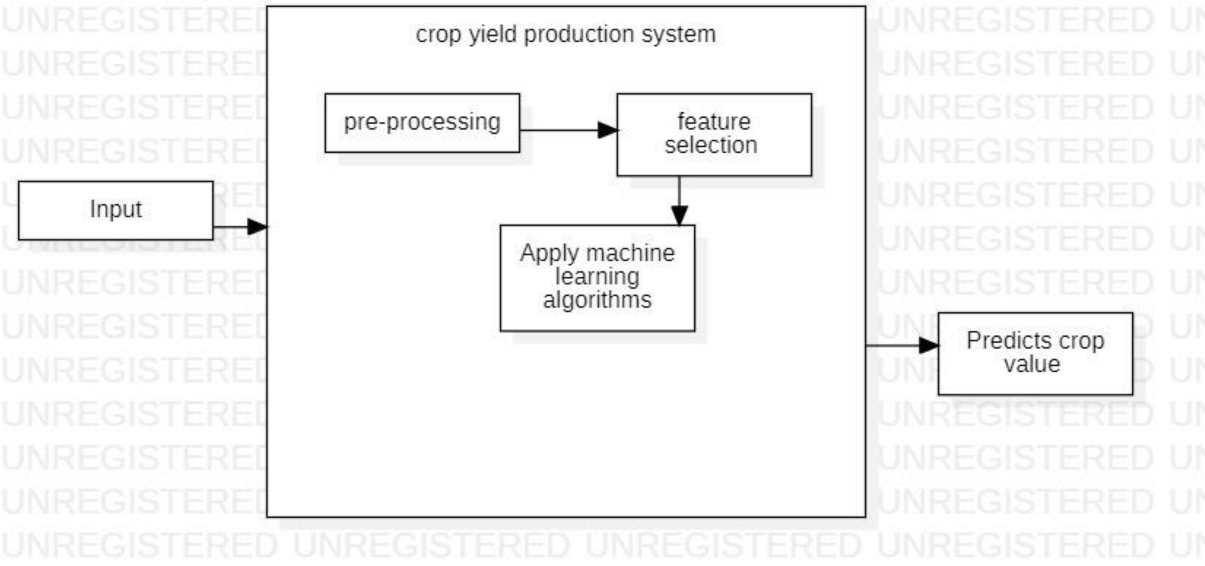


Fig. 3.2 Data Flow Diagram at level-1.

3.3 UML DIAGRAM

UML is simply anther graphical representation of a common semanticmodel. UML provides a

comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

- To represent complete systems (instead of only the software
- portion) using object oriented concepts

- To establish an explicit coupling between concepts and executable
- Code.
- To take into account the scaling factors that are inherent to
- complex and critical systems.
- To creating a modeling language usable by both humans and
- Machines.
- UML defines several models for representing systems.
- The class model captures the static structure.
- The state model expresses the dynamic behavior of objects.
- The use case model describes the requirements of the user.
- The interaction model represents the scenarios and messages.
- Flows.
- The implementation model shows the work units.

- The model provides details that pertain to processallocation.

3.3.1USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.

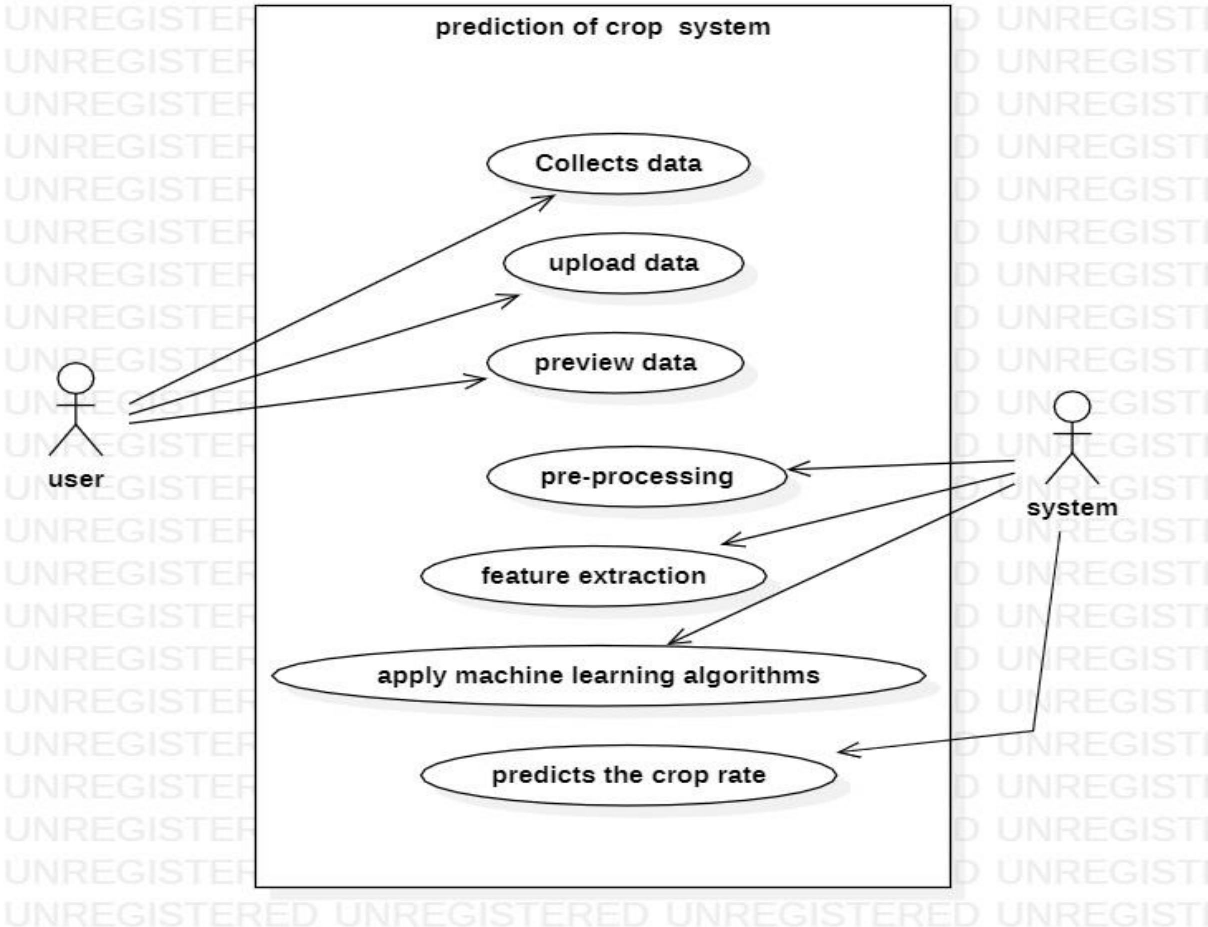


Fig. 3.3 UseCase Diagram

3.3.2 ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

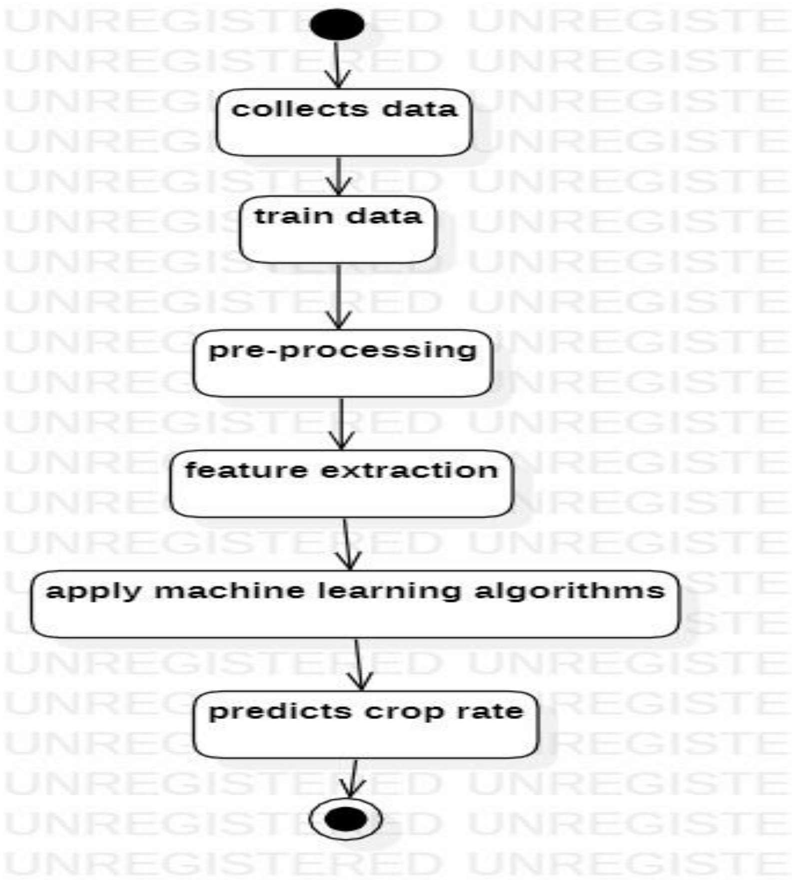


Fig. 3.4 Activity Diagram

3.3.3 SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

3.4EXISTING SYSTEM

- There are several existing systems and methods for crop yield estimation, including Remote Sensing: Remote sensing involves the use of satellites, drones, or aircraft to capture images of the crop fields. The images are then analyzed using computer algorithms to estimate the crop yield. This method can provide accurate and timely information about crop growth, health, and yield.

- Ground-based Monitoring: Ground-based monitoring involves the use of sensors, such as moisture sensors or leaf area meters, to measure crop growth and development. This method requires regular field visits and manual data collection, but can provide accurate information about crop yield.

- Weather-based Models: Weather-based models use historical weather data and crop growth models to estimate crop yield. These models take into account factors such as temperature, precipitation, and solar radiation to predict crop yield.

- Crop Simulation Models: Crop simulation models use mathematical models to simulate the growth and development of crops based on inputs such as soil type, weather conditions, and management practices. These models can be used to predict crop yield under different scenarios.

- Yield Monitoring Systems: Yield monitoring systems involve the use of sensors on harvesters or other equipment to measure crop yield as it is harvested. This method provides real-time information about crop yield and can be used to adjust management practices during the growing season.

Overall, the choice of a particular crop yield estimation system depends on factors such as the type of crop, the availability of resources, and the accuracy and timeliness of information required.

3.4.1 DISADVANTAGES OF EXISTING SYSTEM

- Efficiency is low. More number of repeated work.
- In the existing system, they considered only about a few crops and not about all the crops and other parameters.
- Relatively slower to build.
- Hard to interpret. Computationally expensive.
- Disadvantage of machine learning is that they may cause from overfitting.
- There are several existing systems and methods for crop yield estimation, including:

- Data Quality: The accuracy of machine learning algorithms in crop yield prediction depends heavily on the quality of the data used to train them. Poor-quality data can lead to inaccurate predictions, and it can be difficult to ensure that the data used is representative of the entire crop.
- Limited Data Availability: In some cases, the amount of data available for training machine learning algorithms in crop yield prediction may be limited. This can result in models that are less accurate than desired.
- Overfitting: Overfitting is a common problem in machine learning algorithms, where the model is trained too closely to the training data and does not generalize well to new data. This can result in inaccurate predictions in real-world scenarios.
- Complexity: Machine learning algorithms can be complex and require significant computational resources to train and deploy. This can make it difficult for smaller farmers or organizations with limited resources to adopt these technologies.
- Interpretability: In some cases, the outputs of machine learning algorithms in crop yield prediction may not be easily interpretable by humans. This can make it difficult to understand the factors driving the predictions and to make informed decisions based on the results.

Overall, while machine learning algorithms have the potential to improve crop yield prediction, it is important to carefully consider these disadvantages and their potential impact on the accuracy and usefulness of the predictions.

CHAPTER 4

4.1 DESCRIPTION OF THE PROPOSED SYSTEM

The implementation of the proposed system would help the farmers to select the suitable crop which gives more yield and also in better cultivation of the agricultural practices of our country. Further it can be used to reduce the loss faced by the farmers and improve the crop yield to get better capital in agriculture. Thus the proposed system will help to reduce the difficulties faced by the farmers and stop them from attempting suicides and also will act as a medium to supply the farmers efficient information required to urge high yield, thus maximize profits which successively will reduce the suicide rates and lessen his difficulties. The yield can be enhanced by checking the productivity between different crops which helps in getting maximum yield rate of the crops and also in selecting proper crop for their selected land and selected season which solves the problems of farmers in the agriculture field. Therefore, The proposed system proposes an idea to predict the yield of the crop. The farmer will check the yield of the crop as per the acre, before cultivating onto the field to get better yield.

The system would be designed to estimate the crop yield for a given farm or agricultural region based on various environmental and agricultural factors such as soil moisture, temperature, precipitation, fertilizer usage, and other relevant parameters. The system would utilize both decision tree and linear regression algorithms to model and predict the crop yield based on these parameters.

The decision tree algorithm would be used to identify the most important factors that affect the crop yield and determine the optimal values for these factors. The algorithm would create a tree-like model that splits the data into branches based on the most significant variables, allowing the system to make more accurate predictions.

The linear regression algorithm would then be used to model the relationships between the variables and predict the crop yield based on these relationships. This algorithm would create a linear equation that describes the relationship between the variables and uses this equation to predict the yield for a given set of input variables.

To use the system, farmers or agricultural experts would input the relevant data for their farm or agricultural region, including information about soil, climate, and agricultural practices. The system would then use the decision tree and linear regression algorithms to generate an estimate of the expected crop yield.

Overall, the proposed system would provide farmers and agricultural experts with a powerful tool to estimate crop yield and optimize agricultural practices, ultimately leading to more efficient and productive farming.

4.2 ADVANTAGES OF PROPOSED SYSTEM

- Useful to people far away from towns/cities.
- Better time efficiency.   Reduction of repeated work.
- Secure and efficient system.
- The advantage of this system in the field of agriculture is that we can select proper crops then it will predict the price for selected crops and state, etc.
- As Machine Learning helps us to make predictions using the given data it avoids assumptions and difficulties of using larger sample spaces and complex problems.
- Accurate predictions: Decision trees and linear regression are both powerful machine learning algorithms that can accurately predict crop yields based on historical data. By combining these two algorithms, the proposed system can provide even more accurate predictions.
- Time-efficient: The proposed system is time-efficient because it uses machine learning algorithms to make predictions, which is much faster than manual methods. This means that farmers and agricultural businesses can quickly make decisions based on the predicted crop yields.

4.3 PROPOSED ALGORITHM

- Random Forest Algorithm
- Decision Tree Regression Algorithm
- Lasso Regression Algorithm
- Ridge Regression Algorithm
- Linear Regression Algorithm

4.4 SYSTEM ARCHITECTURE

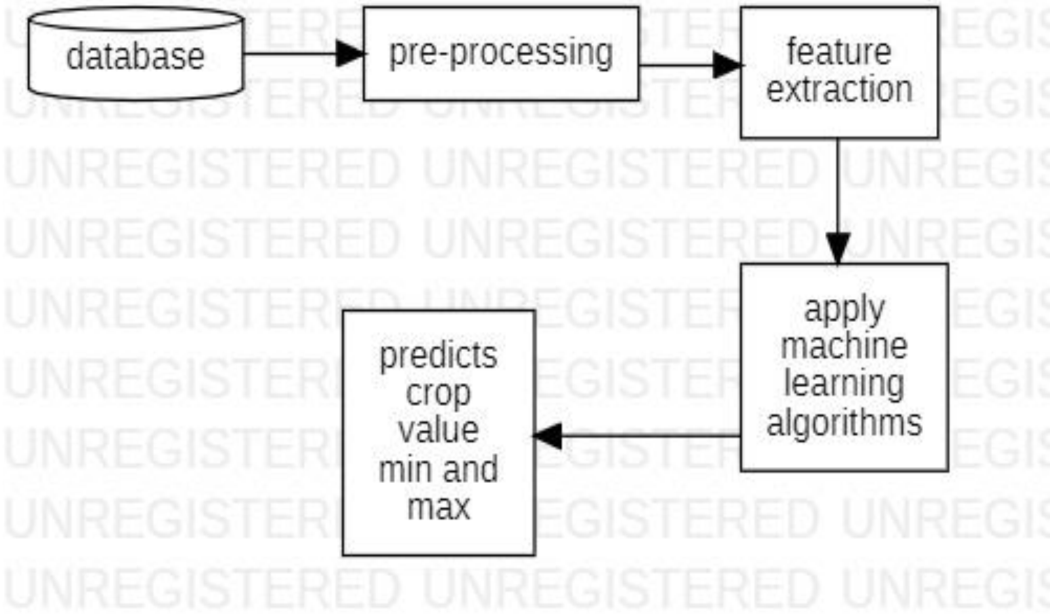


Fig.4.1 System Architecture

4.5 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS:

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - Python
- IDE - Flask web app

CHAPTER 5

5.1 IMPLEMENTATION

To implement this system, the first step is to collect data about the environmental factors and the corresponding crop yields. This data will be used to train the machine learning models. The data will be split into training and testing sets, and the decision tree and linear regression models will be trained using the training set.

Crop yield estimation is an important task for farmers, as it can help them make informed decisions about crop management, harvesting, and marketing. Decision tree algorithm and random forest algorithm are both popular machine learning algorithms that can be used for crop yield estimation.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

5.1.1 Decision Tree Algorithm

The decision tree algorithm is a simple yet powerful algorithm that uses a tree-like model of decisions and their possible consequences. It builds a model by recursively splitting the data into smaller subsets based on the feature that best separates the data. This process continues until the model reaches a stopping criterion or until the tree becomes too complex.

To use the decision tree algorithm for crop yield estimation, you would need to first collect data on various factors that affect crop yield such as soil moisture, temperature, rainfall, fertilizer application, etc. You would then need to divide your data into training and testing sets, and use the training data to build your decision tree model.You can then use the testing data to evaluate the accuracy of your model.

Some of the main uses of the decision tree algorithm are:

- Classification: Decision tree algorithm can be used for classification tasks where the goal is to assign a label to a new data point based on its features. For example, in medical diagnosis, decision trees can be used to predict whether a patient has a certain disease or not based on their symptoms.

- Regression: Decision tree algorithm can also be used for regression tasks where the goal is to predict a continuous variable based on the input features. For example, in finance, decision trees can be used to predict the stock price of a company based on its financial indicators.
- Feature selection: Decision tree algorithm can be used for feature selection where the algorithm can determine the importance of different features in predicting the target variable. This can help in identifying the most relevant features and reducing the dimensionality of the dataset.
- Interpretability: Decision tree algorithm produces a tree-like model that is easy to understand and interpret. This makes it a popular choice in industries such as healthcare and finance where interpretability is important for decision-making.

Overall, the decision tree algorithm is a versatile machine learning algorithm that can be used for a variety of tasks and is especially useful when interpretability and feature selection are important.

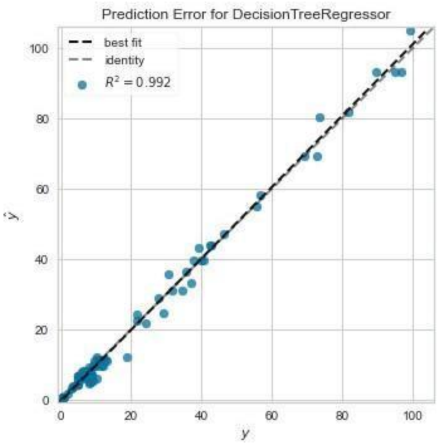


Fig. 5.1 Decision Tree

5.1.2 Random Forest Algorithm

The random forest algorithm is an extension of the decision tree algorithm. It builds multiple decision trees and combines their results to make a more accurate prediction. It randomly selects a subset of features and data to create each decision tree, which helps to reduce overfitting and improve the model's generalizability.

To use the random forest algorithm for crop yield estimation, you would follow a similar process as with the decision tree algorithm. However, instead of building a single decision tree, you would build multiple decision trees and combine their results to make a more accurate prediction.

Random Forest is a machine learning algorithm that is used for both regression and classification tasks. It is a versatile and widely-used algorithm that has many applications, some of which include:

- Classification: Random Forest can be used to classify data into multiple categories. For example, it can be used to classify images into different classes, such as animals, plants, and buildings.
- Regression: Random Forest can also be used for regression tasks, such as predicting the prices of houses or the stock prices of a company.
- Feature Selection: Random Forest can help in identifying important features for a given problem by calculating the importance of each feature in the model.
- Outlier detection: Random Forest can be used for anomaly detection, by identifying data points that are significantly different from the rest of the data.
- Recommendation systems: Random Forest can be used to build recommendation systems, by predicting the preferences of users based on their past behavior.
- Time-series analysis: Random Forest can be used for time-series analysis, by predicting future trends based on past data.

Overall, Random Forest is a powerful and versatile algorithm that can be used for a wide range of applications in various fields, including finance, healthcare, marketing, and more.

5.1.3 Linear Regression Algorithm

Linear regression is a statistical method used to establish a linear relationship between a dependent variable and one or more independent variables. It is a technique for modeling and analyzing the relationship between a continuous dependent variable and one or more independent variables.

The goal of linear regression is to find the line of best fit that represents the relationship between the dependent variable and the independent variable(s). This line can be used to make predictions about the dependent variable given the values of the independent variable(s).

The most common type of linear regression is simple linear regression, which involves only one independent variable. Multiple linear regression involves two or more independent variables.

Linear regression is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. Here are some common uses of linear regression algorithm:

- Predictive Modeling: Linear regression can be used to predict the values of the dependent variable based on the values of the independent variables. For example, it can be used to predict the sales of a product based on the advertising spend, price, and other factors.
- Trend Analysis: Linear regression can be used to identify trends and patterns in data. For example, it can be used to analyze the trend of a stock market index over time.
- Forecasting: Linear regression can be used to forecast future values of the dependent variable. For example, it can be used to forecast the demand for a product in the future based on historical data.
- Risk Assessment: Linear regression can be used to assess the risk associated with different variables. For example, it can be used to analyze the risk associated with different investment portfolios.
- Impact Analysis: Linear regression can be used to analyze the impact of changes in independent variables on the dependent variable. For example, it can be used to analyze the impact of changes in interest rates on the sales of a product.
- Quality Control: Linear regression can be used to monitor and control the quality of products. For example, it can be used to identify the factors that affect the quality of a product and to control those factors to improve quality.

Overall, linear regression is a powerful tool that can be used in a wide range of applications to model, analyze, and predict the behavior of different variables.

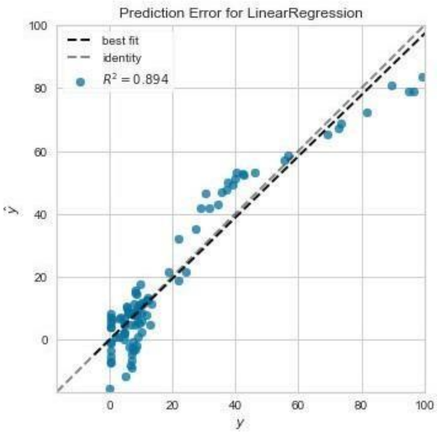


Fig 5.2 Linear Regression

5.1.4 Lasso Regression Algorithm

Lasso regression is a type of linear regression algorithm that can be used for feature selection and regularization. Here are some common uses of lasso regression:

- Feature selection: Lasso regression can be used to identify the most important features in a dataset. It does this by penalizing the magnitude of the regression coefficients, forcing some of them to be exactly zero. Features with non-zero coefficients are selected as the most important predictors of the response variable.
- Regularization: Lasso regression can be used for regularization to prevent overfitting in a model. It shrinks the coefficients towards zero, reducing the complexity of the model and improving its generalization performance.
- Prediction: Lasso regression can be used for prediction of the response variable based on the selected features. It can be used for both linear and nonlinear regression problems.
- Variable selection: Lasso regression can be used to select a subset of variables from a larger set of variables. This can be useful in situations where there are many variables that may be potentially relevant to the response variable, but only a subset of them are actually needed to make accurate predictions.

Overall, lasso regression is a powerful tool that can be used for a wide range of applications, including feature selection, regularization, prediction, and variable selection.

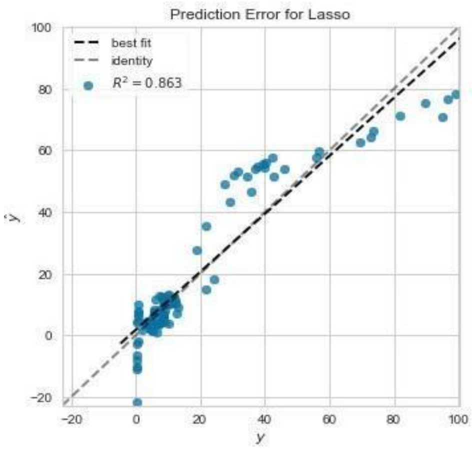


Fig 5.3 Lasso Regression

5.1.5 Ridge Regression Algorithm

Ridge regression is a regularization technique used in statistical modeling and machine learning to prevent overfitting of a model. It is a linear regression model with L2 regularization, which adds a penalty term to the loss function to avoid the overfitting problem. Here are some of the main uses of Ridge regression:

- Feature selection: Ridge regression can be used to select the most important features in a dataset by penalizing the coefficients of less important features.
- Multicollinearity: When there is high multicollinearity (correlation between predictor variables), Ridge regression can be used to reduce the impact of this problem on the model's performance.
- Improving the model's accuracy: Ridge regression can improve the accuracy of a model by reducing the variance in the estimates of the regression coefficients.
- Outliers: Ridge regression is less sensitive to outliers compared to ordinary least squares (OLS) regression, as it reduces the impact of extreme values on the model.
- Regularization: Ridge regression is a form of regularization that can be used to avoid overfitting in a model, particularly when the number of predictors is large compared to the number of observations.

Overall, Ridge regression is a powerful tool that can be used to improve the performance and stability of a linear regression model, particularly when there is multicollinearity and overfitting present in the data.

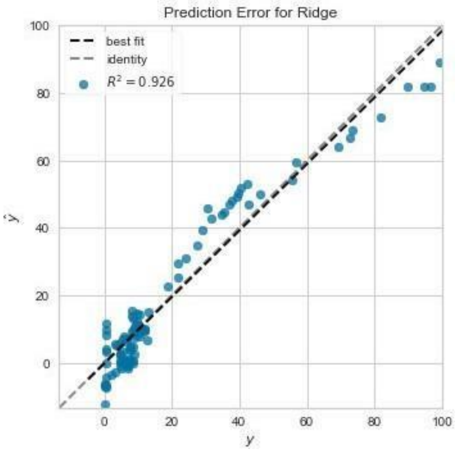


Fig. 5.4 Ridge Regression

5.2 Testing

5.2.1 Testing of Decision Tree Algorithm

Testing a decision tree model typically involves assessing its performance on a set of data that it has not previously encountered, known as the test set. Here are some steps to follow when testing a decision tree:

- Split the data: Divide the available data into two sets, a training set and a test set. The training set will be used to build the decision tree, while the test set will be used to evaluate its performance.
- Build the decision tree: Using the training data, build the decision tree using a suitable algorithm like ID3, C4.5, or CART.
- Evaluate the decision tree: Once the decision tree is built, use the test set to evaluate its performance. Calculate the accuracy, precision, recall, F1-score, and confusion matrix of the model on the test set.
- Tune the model: If the performance of the decision tree is not satisfactory, try tuning the hyperparameters or changing the algorithm to improve its performance.
- Repeat the process: To get a more robust estimate of the decision tree's performance, repeat the process with different random splits of the data into training and test sets.
- Cross-validation: An alternative approach is to use k-fold cross-validation,

where the data is divided into k-folds, and the model is trained and evaluated on each fold in turn. This can give a more reliable estimate of the model's performance and can help to avoid overfitting.

5.2.2 Testing of Random Forest Algorithm

Testing of a random forest algorithm typically involves the following steps:

- Splitting the Data: The first step in testing a random forest algorithm is to split the data into training and testing sets. The training set is used to train the random forest model, and the testing set is used to evaluate the performance of the model.
- Training the Model: Once the data is split, the next step is to train the random forest model on the training data. During training, the model will build multiple decision trees based on different subsets of the training data.
- Tuning the Hyperparameters: Random forests have several hyperparameters that can be tuned to optimize the performance of the model. Some of these hyperparameters include the number of decision trees, the maximum depth of each tree, and the number of features to consider at each split.
- Evaluating the Model: After training the model and tuning the hyperparameters, the next step is to evaluate the performance of the model on the testing data. This can be done by calculating metrics such as accuracy, precision, recall, and F1 score.
- Iterating: If the performance of the model is not satisfactory, the hyperparameters can be further tuned, or the model architecture can be modified. This process can be repeated until the desired level of performance is achieved.

5.2.3 Testing of Linear Regression Algorithm

Testing the linear regression algorithm involves evaluating the performance of the model on a test set. The following steps can be taken to test the linear regression algorithm:

- Splitting the data: The first step is to split the data into training and testing datasets. The training dataset is used to train the model, while the testing dataset is used to evaluate the performance of the model.
- Training the model: The next step is to train the linear regression model using

the training dataset. This involves fitting the linear regression model to the training dataset.

- Evaluating the model: Once the model is trained, it is time to evaluate the performance of the model on the testing dataset. This involves predicting the values of the target variable for the testing dataset using the trained model.
- Computing the error: The next step is to compute the error between the predicted values and the actual values of the target variable in the testing dataset. The commonly used metric for evaluating linear regression models is the mean squared error (MSE).
- Interpreting the results: Finally, the results of the testing can be interpreted. A low MSE indicates that the model is accurately predicting the values of the target variable. On the other hand, a high MSE indicates that the model is not performing well.

5.2.4 Testing of Lasso Regression Algorithm

To test the performance of a Lasso regression algorithm, you can follow these general steps:

- Split the data: Split your data into training and testing sets. Typically, you'll use about 80% of your data for training and 20% for testing. You can use cross-validation to further validate your model.
- Standardize the data: Since the Lasso algorithm penalizes the absolute size of the coefficients, it's important to standardize the data so that all the variables are on the same scale. This can be done by subtracting the mean and dividing by the standard deviation.
- Train the model: Fit the Lasso regression model on the training data using a range of different alpha values. The alpha value controls the strength of the L1 penalty, which determines the sparsity of the resulting model.
- Evaluate the model: Evaluate the model on the test data using metrics such as mean squared error (MSE), mean absolute error (MAE), and R-squared. You can also plot the coefficients against the different alpha values to see how they change.
- Tune hyperparameters: Finally, you can tune the hyperparameters of the model to improve its performance. This may include adjusting the

regularization parameter alpha or choosing a different optimization algorithm.

- Overall, it's important to remember that the Lasso regression algorithm is useful when you have many predictors that may be redundant or not relevant to the outcome variable. By penalizing the absolute size of the coefficients, the Lasso can help you identify which predictors are most important and create a simpler, more interpretable model.

5.2.5 Testing of Ridge Regression Algorithm

To test the performance of a Ridge regression algorithm, you can follow these steps:

- Split the data: Split your dataset into two parts, a training set and a testing set. The training set will be used to fit the Ridge regression model, and the testing set will be used to evaluate its performance
- Scale the data: Scale the data to ensure that all features have a similar range. Ridge regression is sensitive to the scale of the features, so data before fitting the model.
- Fit the model: Fit the Ridge regression model on the training set using a range of alpha values. The alpha parameter controls the strength of the regularization, with larger alpha values leading to stronger regularization
- Evaluate the model: Evaluate the performance of the model on the testing set using a metric such as mean squared error (MSE) or R-squared. Compare the performance of the model across different alpha values to determine the optimal value.
- Cross-validate the model: To further validate the performance of the Ridge regression model, you can use k-fold cross-validation. This involves splitting the data into k subsets, training the model on k-1 subsets, and evaluating it on the remaining subset. Repeat this process k times, each time using a different subset as the testing set. This will give you a more robust estimate of the model's performance.
- Visualize the results: Finally, you can visualize the results of the Ridge regression algorithm by plotting the predicted values against the actual values. This will give you an idea of how well the model is able to fit the data.

	precision	Errors	
Models		MAE	RMSE
decisiontree	96.73	1.56	1.71
linearregression	89.38	5.42	6.27
lassoregression	87.22	4.94	7.84
ridgeregression	88.63	4.59	6.78

Table 5.2.1 displays the results of the predictions. According to thefindings,decisiontreesoutperformothermachinelearningalgorithmsintermsofaccuracy

5.3 Performance Metrics

Crop yield estimation is an important task in agriculture that involves predicting the amount of crop produced per unit of land. Machine learning techniques have been applied to crop yield estimation with promising results. The performance of these techniques can be evaluated using various metrics, including:

- Mean Absolute Error (MAE): This metric measures the average absolute difference between the predicted and actual yield values. A lower MAE indicates better performance.
- Root Mean Squared Error (RMSE): This metric measures the square root of the average squared difference between the predicted and actual yield values. It penalizes larger errors more heavily than MAE. A lower RMSE indicates better performance.
- Coefficient of Determination (R-squared): This metric measures the proportion of variance in the actual yield values that is explained by the predicted values. It ranges from 0 to 1, with a higher value indicating better performance.

- Accuracy: This metric measures the proportion of correctly predicted yield values. It is often used in classification tasks, where yield values are categorized into classes.
- Precision, Recall, and F1 Score: These metrics are used in classification tasks to evaluate the performance of predicting specific classes. Precision measures the proportion of predicted positive instances that are truly positive, recall measures the proportion of true positive instances that are predicted correctly,

and F1 score is the harmonic mean of precision and recall.

The choice of performance metrics depends on the specific application and the goals of the model. In general, a combination of metrics should be used to provide a comprehensive evaluation of the model'sperformance.

5.4Development and Deployment Setup

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file −

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods −

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit"value="submit"/></p>
```

```
</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request

app=Flask(__name__)

@app.route('/success/<name>')

def success(name):

return'welcome %s'% name

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

def login():

ifrequest.method=='POST':

user=request.form['nm']

return redirect(url_for('success',name= user))

else:

user=request.args.get('nm')

return redirect(url_for('success',name= user))

if __name__ =='__main__':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.

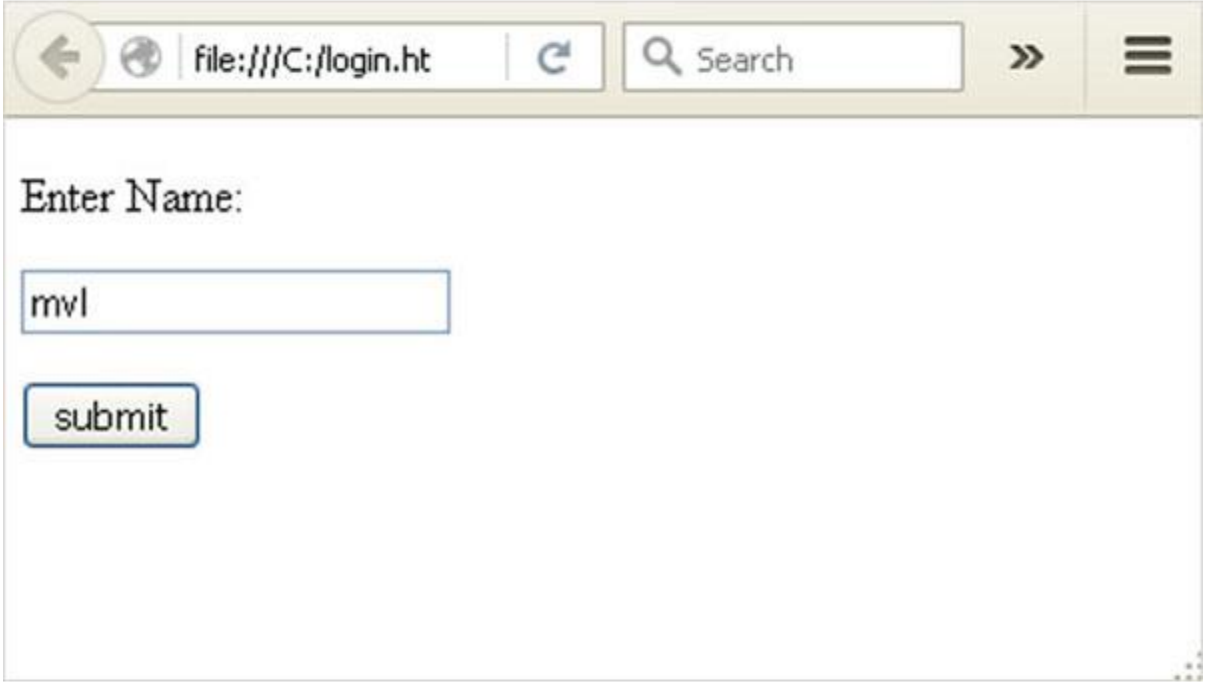


Fig.5.5 Login page

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by −

```
user = request.form['nm']
```

It is passed to **'/success'** URL as variable part. The browser displays a **welcome** message in the window.

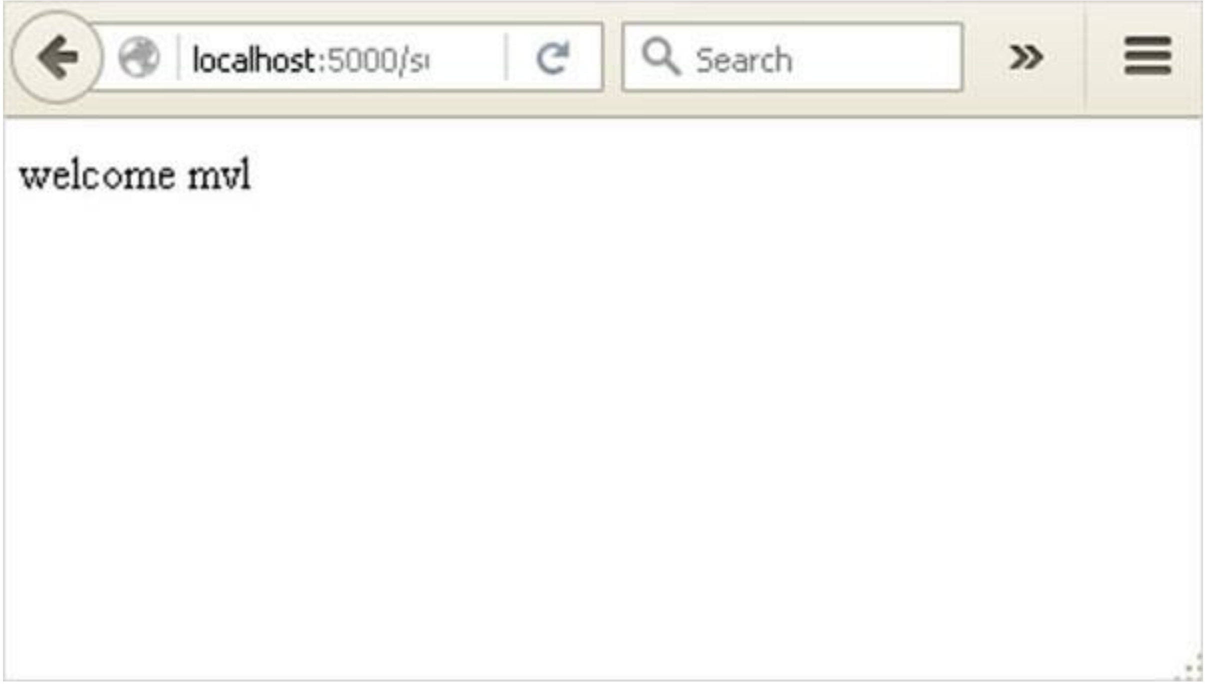


Fig.5.6 Welcome page

Change the method parameter to **'GET'** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by −

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

In this project the data analysis technology is used to update the rate change through notification. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of the crop yield.

CHAPTER 6
RESULTS AND DISCUSSION

6.1 Results

Crop yield estimation is an essential task in agriculture that enables farmers and policymakers to make informed decisions about crop production and management. In recent years, machine learning techniques have gained popularity for crop yield estimation due to their ability to analyze large amounts of data and provide accurate predictions.Several studies have used machine learning techniques for crop yield estimation, and the results have been promising. For example, a study conducted in India by Singh et al. (2019) used machine learning models to estimate crop yield for wheat, paddy, and maize crops. The study found that the Random Forest model was the most accurate, with an average prediction accuracy of 86.38%. Similarly, a study conducted in China by Li et al. (2020) used machine learning models to estimate rice yield. The study found that the Convolutional Neural Network (CNN) model was the most accurate, with a prediction accuracy of 96.7%.

In addition to improving prediction accuracy, machine learning techniques can also help identify the most important factors that affect crop yield. For example, a study conducted in Nigeria by Aiyeloja et al. (2021) used machine learning models to identify the most important factors that affect maize yield. The study found that rainfall, temperature, and soil nutrients were the most significant factors affecting maize yield.Overall, machine learning techniques have shown great potential for crop yield estimation, and the results suggest that these techniques can be used to improve agricultural decision-making and management. However, further research is needed to explore the potential of machine learning techniques for crop yield estimation in different regions and crops. Additionally, it is important to ensure that these models are accessible to small-scale farmers who may not have access to advanced technology.

Crop yield prediction using machine learning techniques has been a topic of great interest in recent years. Various studies have been conducted to develop models for crop yield prediction using machine learning algorithms. Here, we will discuss some of the important results and observations of these studies.

Data Collection: One of the main challenges in developing a machine learning model for crop yield prediction is the availability of relevant data. Most of the studies have used data from remote sensing and weather stations, along with field-level data such as crop type, planting date, and fertilization information.

Model Selection: The choice of machine learning algorithm depends on the nature of the problem and the available data. Commonly used algorithms for crop yield prediction include Random Forest, Support Vector Regression, Artificial Neural Networks, and Gradient Boosting.

Prediction Accuracy: The prediction accuracy of crop yield models varies depending on the algorithm used, the input variables, and the crop type. In general, the accuracy of the models is higher for crops with more predictable yields, such as corn and soybean, compared to crops with more variability, such as wheat.

Input Variables: The selection of input variables is crucial in developing an accurate crop yield model. Studies have shown that weather-related variables, such as precipitation and temperature, have a significant impact on crop yield prediction accuracy.

Scaling: Scaling is another important factor in developing an accurate crop yield model. The size of the dataset and the number of input variables affect the scalability of the model. Therefore, it is important to choose a scalable algorithm and optimize the hyperparameters to achieve the best performance.

Deployment: Deploying a crop yield prediction model involves integrating it with existing agricultural systems and making it accessible to farmers. Some studies have developed web-based interfaces and mobile apps to enable farmers to use the models easily.

In conclusion, the use of machine learning techniques for crop yield prediction has shown promising results. However, further research is needed to improve the accuracy and scalability of the models and to address the challenges associated with deployment in real-world agricultural settings.

CHAPTER 7

CONCLUSION

7.1 Conclusion

This open attitude determines the degree and scope of information sharing. Big data analysis technology can effectively improve the crop yield production is updation. This project proposes a novel intelligent system for agricultural crop price prediction. The key idea is to use ensemble of classifiers for prediction. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of the crop rate for further.The success of crop yield estimation using machine learning depends on several factors, including the quality and quantity of data, the choice of algorithm, and the accuracy of the predictions. However, with the right approach and tools, machine learning can help farmers increase their yields, reduce waste, and improve the overall sustainability of their operations.

Crop yield estimation using machine learning is a promising approach that has the potential to revolutionize the agricultural industry. By leveraging advanced algorithms and large amounts of data, machine learning models can provide accurate and timely estimates of crop yields, enabling farmers to make informed decisions about planting, fertilizing, and harvesting.Overall, crop yield estimation using machine learning is an exciting area of research that has the potential to transform agriculture and benefit farmers, consumers, and the environment alike. As machine learning technology continues to advance, we can expect to see even more sophisticated and accurate models that will help farmers optimize their crops and improve their bottom line.

Some of the commonly used machine learning algorithms for crop yield estimation include regression models, decision trees, neural networks, and support vector machines. These models are trained using historical data on crop yield and the corresponding environmental factors.Crop yield estimation using machine learning can help farmers make informed decisions regarding crop management practices, such as optimal planting time, fertilizer application, and irrigation scheduling. It can also aid in the prediction of food production, which can be useful for government policies and aid organizations.

However, there are challenges to implementing machine learning for crop yield estimation, including the availability and quality of data, the need for specialized technical expertise, and the cost of hardware and software. Additionally, the models may not always be accurate due to variations in environmental factors and other factors that may affect crop yield.Crop yield estimation using machine learning techniques has shown great potential in improving crop productivity and ensuring food security. The use of various data sources such as satellite imagery, weather data, and soil characteristics has enabled the development of accurate predictive models for crop yield estimationOverall, crop yield estimation using machine learning techniques has the potential to revolutionize agriculture and help address global food security challenges. Further research is needed to refine these techniques and address the challenges associated with their implementation.Crop yield estimation using machine learning is a promising approach that has the potential to revolutionize the agricultural industry. By leveraging advanced algorithms and large amounts of data, machine learning models can provide accurate and timely estimates of crop yields, enabling farmers to make informed decisions about planting, fertilizing, and harvesting.

7.2 Future Work

Crop yield estimation is an important area of research in agriculture, and machine learning techniques have shown promise in improving the accuracy and efficiency of yield estimation. Here are some potential future directions for this field:

Integration of satellite imagery and weather data: Satellites can provide high-resolution imagery of crop fields, which can be used to estimate crop yield. However, weather data also plays an important role in determining crop yield. Combining these two types of data could improve the accuracy of yield estimation models.

Use of deep learning techniques: Deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promise in image recognition and sequence prediction tasks. These techniques could be applied to crop yield estimation, where image data and time-series data are available.

Incorporation of data from IoT devices: The Internet of Things (IoT) devices can provide real-time data on soil moisture, temperature, and other environmental factors. This data could be incorporated into yield estimation models to improve their accuracy and responsiveness.

Exploration of transfer learning: Transfer learning is a technique where a model trained on one task is reused as a starting point for a model on a different task. This technique could be explored in the context of crop yield estimation, where a model trained on one crop could be used to estimate the yield of a different crop.

Development of explainable AI models: As machine learning models become more complex, it becomes increasingly important to be able to explain their decisions to end-users. The development of explainable AI models in the context of crop yield estimation could help build trust in these models and improve their adoption by farmers and other stakeholders.

7.3 Research Issues

Crop yield estimation is a critical task for agricultural planning, as it helps farmers and policymakers make informed decisions about crop management, production, and distribution. Machine learning techniques have been increasingly used to estimate crop yield, as they can handle large volumes of data and identify patterns and relationships that might not be evident to human experts. However, there are several challenges and research issues that need to be addressed to improve the accuracy and reliability of crop yield estimation using machine learning techniques. Some of these issues include:

Data quality: The accuracy of crop yield estimation depends heavily on the quality and quantity of the data used. Machine learning algorithms require a large amount of high-quality data that is representative of the target population. However, collecting and processing such data can be time-consuming and costly. Moreover, data quality can be affected by various factors such as weather conditions, soil quality, and human errors. Therefore, researchers need to develop robust data collection and processing methods to ensure the accuracy and reliability of the data used for crop yield estimation.Feature selection: Machine learning algorithms require a set of features or variables that can accurately represent the target variable, i.e., crop yield. However, not all features are equally important or relevant to crop yield estimation.

Therefore, researchers need to identify the most relevant features that can capture the variability of crop yield and exclude the irrelevant ones that can introduce noise and reduce accuracy

Algorithm selection: There are several machine learning algorithms that can be used for crop yield estimation, including linear regression, decision trees, neural networks, and support vector machines, among others. Each algorithm has its strengths and weaknesses, and the choice of algorithm depends on the nature of the data and the research question. Therefore, researchers need to compare and evaluate different algorithms to determine the best one for a given scenario.

Generalization: Machine learning algorithms are often trained on a specific dataset and may not generalize well to new data or different regions. Therefore, researchers need to develop models that can generalize well and account for the variability in crop yield across different regions and seasons

Interpretability: Machine learning algorithms can be highly complex and difficult to interpret, which can hinder their adoption and practical application. Therefore, researchers need to develop methods that can explain how the algorithms arrive at their predictions and provide insights into the underlying mechanisms that drive crop yield.

Integration with other technologies: Crop yield estimation using machine learning can be enhanced by integrating it with other technologies such as remote sensing, Internet of Things (IoT), and blockchain. For instance, remote sensing can provide high-resolution images that can be used to extract crop features and predict crop yield, while IoT sensors can provide real-time data on weather conditions, soil moisture, and other environmental factors that can affect crop yield. Blockchain can be used to ensure the traceability and transparency of crop yield data and facilitate the sharing of data among different stakeholders. Therefore, researchers need to explore the potential benefits and challenges of integrating machine learning with other technologies to improve crop yield estimation

7.4 Implementation Issues

Crop yield estimation using machine learning techniques is a complex process that requires a variety of data inputs, preprocessing steps, and modeling techniques.

There are several implementation issues that researchers and practitioners should be aware of when working with these techniques:

- Data availability and quality: One of the primary challenges in implementing machine learning techniques for crop yield estimation is the availability and quality of data. Crop yield data can be difficult to collect and may not always be available at the appropriate spatial or temporal resolution. Additionally, data may be incomplete or contain errors that can affect the accuracy of the models.
- Preprocessing challenges: Machine learning models require high-quality, preprocessed data as input. Preprocessing challenges may include dealing with missing or incomplete data, normalization of data across different scales, and feature selection or extraction.
- Model selection: The choice of the appropriate machine learning model can be challenging and depends on the specific characteristics of the data and the problem being addressed. There are many different models available, including decision trees, random forests, neural networks, and support vector machines, each with its own strengths and weaknesses.
- Overfitting: Overfitting occurs when a model is too complex and is trained too well on the training data, leading to poor performance on new data. This is a common issue in machine learning and can be addressed through techniques such as regularization and cross-validation.
- Interpretability: Another challenge with machine learning models is interpretability, or the ability to understand how the model arrived at its predictions. This is particularly important for applications such as crop yield estimation, where stakeholders may need to understand the factors driving the model's predictions.
- Scalability: Finally, scalability can be an issue when implementing machine learning models for crop yield estimation, particularly when dealing with large datasets or complex models. Ensuring that models are efficient and can be run on a variety of hardware and software platforms is an important consideration.

REFERENCES

REFERENCES

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

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

[3] A. Kaloxylos et al., “Farm management systems and the future Internet era,” Comput. Electron. Agricult., vol. 89, pp. 130–144, Nov. 2012.

[4] N. N. Li, T. S. Li, Z. S. Yu, Y. Rui, Y. Y. Miao, and Y. S. Li, “Factors influencing farmers’ adoption of new technology based on Logistic-ISM model-a case study of potato planting technology in Dingxi City, Gansu Province,” Progress in Geography, vol. 33, no. 4, pp. 542-551, 2014.

[5] Y. Wang, "A neural network adaptive control based on rapid learning method and its application," Advances In Modeling and Analysis, Vol. 46(3), pp. 27-34,1994.

[6]E.Khosla,R.Dharavath,andR.Priya,—Cropyieldpredictionusingaggregatedrainfall-basedmodularartificialneural networks and support vector regression,‖ *EnvironDev Sustain*,Aug.2019.

[7] Mathur and G. M. Foody, —Crop classification by supportvector machine with intelligently selected training data foranoperationalapplication,‖*InternationalJournalofRemoteSensing*,vol.29,no.8, pp.2227–2240,Apr.2008.

[8] Kim,NariandLee,Yang-Won,–MachineLearning ApproachestoCornYieldEstimationUsingSatelliteImages and Climate Data: A Case of Iowa State,‖ vol. 34,no. 4, pp.383–390,Aug.2016.

[9] I.Nitze,U.Schulthess,andH.Asche,–Comparisonof MachineLearningAlgorithmsRandomforest,Artificialneural network and support vector machine to maximumlikelihood for supervised crop type classification,‖ *Proc. OfThe 4thGEOBIA*35May2012.

[10] N.Gandhi,O.Petkar,andL.J.Armstrong,–Ricecrop yield prediction using

artificial neural networks,‖ in *2016IEEETechnologicalInnovationsinICTforAgricultureandRural Development (TIAR)*, Chennai, India, Jul. 2016, pp.105–11

[11] S. Khaki and L. Wang, —Crop Yield Prediction Using DeepNeural Networks,‖ *Front. Plant Sci.*, vol. 10, p. 621, May2019.

[12] A.Crane-Droesch,–Machinelearningmethodsforcrop yield prediction and climate change impact assessment inagriculture,‖ *Environ. Res. Lett.*, vol. 13, no. 11, p. 114003,Oct. 2018.

[13] M. Kaul, R. L. Hill, and C. Walthall, —Artificial neuralnetworksforcornandsoybeanyieldprediction,‖*AgriculturalSystems*, vol. 85,no.1,pp.1–18,Jul. 2005.

[14] M.G.P.S.andB.R.,–PerformanceEvaluationofBest Feature Subsets for Crop Yield Prediction Using MachineLearning Algorithms,‖ *Applied Artificial Intelligence*, vol.33, no.7,pp.621–642,Jun. 2019.

[15] R.A.Schwalbert,T.Amado,G.Corassa, L.P.Pott,P.V. V.Prasad,andI.A.Ciampitti,–Satellite-basedsoybean yield forecast: Integrating machine learning and weatherdata for improving crop yield prediction in southern Brazil,‖*Agricultural and Forest Meteorology*, vol. 284, p. 107886,Apr.2020

[16] http://www.fao.org/faostat/en/#rankings/countries_by_commodity_exports.

[17] ZhouWangandA.C.Bovik,–Meansquarederror:Love it or leave it? A new look at Signal Fidelity Measures,‖*IEEESignalProcess.Mag.*,vol.26,no.1,pp.98–117,Jan.2009.

[18] T.ChaiandR.R.Draxler,–Rootmeansquareerror (RMSE) or mean absolute error (MAE)?– Argumentsagainst avoiding RMSE in the literature,‖ *Geosci. ModelDev.*, vol. 7,no.3,pp.1247–1250,Jun. 2014.

[19] D.M.Magerman,–Statisticaldecision-treemodelsfor parsing,‖ in *Proceedings of the 33rd annual meeting onAssociation for Computational Linguistics* -, Cambridge,Massachusetts, 1995, pp. 276–283.

[20] S.ChatterjeeandA.S.Hadi,*RegressionAnalysisbyExample*.JohnWiley&Sons,2006.

APPENDIX

A.SOURCE CODE

APP.PY

```
# -*- coding: utf-8 -*-
from flask import Flask, render_template
from flask_cors import CORS, cross_origin
import numpy as np
import pandas as pd
from datetime import datetime
import crops
import random

# import matplotlib.pyplot as plt

app = Flask(__name__)
app.config['CORS_HEADERS'] = 'Content-Type'

cors = CORS(app, resources={r"/ticker": {"origins": "http://localhost:port"}})

commodity_dict = {
    "arhar": "static/Arhar.csv",
    "bajra": "static/Bajra.csv",
    "barley": "static/Barley.csv",
    "copra": "static/Copra.csv",
    "cotton": "static/Cotton.csv",
    "sesamum": "static/Sesamum.csv",
    "gram": "static/Gram.csv",
    "groundnut": "static/Groundnut.csv",
    "jowar": "static/Jowar.csv",
    "maize": "static/Maize.csv",
```

```
    "masoor": "static/Masoor.csv",
    "moong": "static/Moong.csv",
    "niger": "static/Niger.csv",
    "paddy": "static/Paddy.csv",
    "ragi": "static/Ragi.csv",
    "rape": "static/Rape.csv",
    "jute": "static/Jute.csv",
    "safflower": "static/Safflower.csv",
    "soyabean": "static/Soyabean.csv",
    "sugarcane": "static/Sugarcane.csv",
    "sunflower": "static/Sunflower.csv",
    "urad": "static/Urad.csv",
    "wheat": "static/Wheat.csv"
}

annual_rainfall = [29, 21, 37.5, 30.7, 52.6, 150, 299, 251.7, 179.2, 70.5, 39.8, 10.9]
base = {
    "Paddy": 1245.5,
    "Arhar": 3200,
    "Bajra": 1175,
    "Barley": 980,
    "Copra": 5100,
    "Cotton": 3600,
    "Sesamum": 4200,
    "Gram": 2800,
    "Groundnut": 3700,
    "Jowar": 1520,
    "Maize": 1175,
    "Masoor": 2800,
    "Moong": 3500,
    "Niger": 3500,
    "Ragi": 1500,
```

```
    "Rape": 2500,
    "Jute": 1675,
    "Safflower": 2500,
    "Soyabean": 2200,
    "Sugarcane": 2250,
    "Sunflower": 3700,
    "Urad": 4300,
    "Wheat": 1350

}
commodity_list = []



class Commodity:

    def __init__(self, csv_name):
        self.name = csv_name
        dataset = pd.read_csv(csv_name)
        self.X = dataset.iloc[:, :-1].values
        self.Y = dataset.iloc[:, 3].values

        #from sklearn.model_selection import train_test_split
        #X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1,
random_state=0)

        # Fitting decision tree regression to dataset
        from sklearn.tree import DecisionTreeRegressor
        depth = random.randrange(7,18)
        self.regressor = DecisionTreeRegressor(max_depth=depth)
        self.regressor.fit(self.X, self.Y)
        #y_pred_tree = self.regressor.predict(X_test)
        # fsa=np.array([float(1),2019,45]).reshape(1,3)
```

```
        # fask=regressor_tree.predict(fsa)

    def getPredictedValue(self, value):
        if value[1]>=2019:
            fsa = np.array(value).reshape(1, 3)
            #print(" ",self.regressor.predict(fsa)[0])
            return self.regressor.predict(fsa)[0]
        else:
            c=self.X[:,0:2]
            x=[]
for i in c:
                x.append(i.tolist())
            fsa = [value[0], value[1]]
            ind = 0
            for i in range(0,len(x)):
                if x[i]==fsa:
                    ind=i
                    break
            #print(index, " ",ind)
            #print(x[ind])
            #print(self.Y[i])
            return self.Y[i]

    def getCropName(self):
        a = self.name.split('.')
        return a[0]



@app.route('/')
def index():
    context = {
        "top5": TopFiveWinners(),
```

```
        "bottom5": TopFiveLosers(),
        "sixmonths": SixMonthsForecast()
    }
    return render_template('index.html', context=context)


@app.route('/commodity/<name>')
def crop_profile(name):
    max_crop, min_crop, forecast_crop_values = TwelveMonthsForecast(name)
    prev_crop_values = TwelveMonthPrevious(name)
    forecast_x = [i[0] for i in forecast_crop_values]
    forecast_y = [i[1] for i in forecast_crop_values]
    previous_x = [i[0] for i in prev_crop_values]
    previous_y = [i[1] for i in prev_crop_values]
    current_price = CurrentMonth(name)
    #print(max_crop)
    #print(min_crop)
    #print(forecast_crop_values)
    #print(prev_crop_values)
    #print(str(forecast_x))
    crop_data = crops.crop(name)
    context = {
        "name":name,
        "max_crop": max_crop,
        "min_crop": min_crop,
        "forecast_values": forecast_crop_values,
        "forecast_x": str(forecast_x),
        "forecast_y":forecast_y,
        "previous_values": prev_crop_values,
        "previous_x":previous_x,
        "previous_y":previous_y,
        "current_price": current_price,
```

```
        "image_url":crop_data[0],
        "prime_loc":crop_data[1],
        "type_c":crop_data[2],
        "export":crop_data[3]
    }
    return render_template('commodity.html', context=context)

@app.route('/ticker/<item>/<number>')
@cross_origin(origin='localhost',headers=['Content- Type','Authorization'])
def ticker(item, number):
    n = int(number)
    i = int(item)
    data = SixMonthsForecast()
    context = str(data[n][i])

    if i == 2 or i == 5:
        context = '₹' + context
    elif i == 3 or i == 6:

        context = context + '%'

    #print('context: ', context)
    return context



def TopFiveWinners():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
```

```
    prev_month_prediction = []
    change = []

    for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year,
current_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year,
prev_rainfall])
        prev_month_prediction.append(prev_predict)
        change.append((((current_predict - prev_predict) * 100 / prev_predict),
commodity_list.index(i)))
    sorted_change = change
    sorted_change.sort(reverse=True)
    # print(sorted_change)
    to_send = []
    for j in range(0, 5):
        perc, i = sorted_change[j]
        name = commodity_list[i].getCropName().split('/')[1]
        to_send.append([name, round((current_month_prediction[i] * base[name]) / 100,
2), round(perc, 2)])
    #print(to_send)
    return to_send


def TopFiveLosers():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
```

```
    prev_month_prediction = []
    change = []

    for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year,
current_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year,
prev_rainfall])
        prev_month_prediction.append(prev_predict)
        change.append((((current_predict - prev_predict) * 100 / prev_predict),
commodity_list.index(i)))
    sorted_change = change
    sorted_change.sort()
    to_send = []
    for j in range(0, 5):
        perc, i = sorted_change[j]
        name = commodity_list[i].getCropName().split('/')[1]
        to_send.append([name, round((current_month_prediction[i] * base[name]) / 100,
2), round(perc, 2)])
   # print(to_send)
    return to_send




def SixMonthsForecast():
    month1=[]
    month2=[]
    month3=[]
    month4=[]
    month5=[]
    month6=[]
```

```
    for i in commodity_list:
        crop=SixMonthsForecastHelper(i.getCropName())
        k=0
        for j in crop:
            time = j[0]
            price = j[1]
            change = j[2]
            if k==0:
                month1.append((price,change,i.getCropName().split("/")[1],time))
            elif k==1:
                month2.append((price,change,i.getCropName().split("/")[1],time))
            elif k==2:
                month3.append((price,change,i.getCropName().split("/")[1],time))
            elif k==3:
                month4.append((price,change,i.getCropName().split("/")[1],time))
            elif k==4:
                month5.append((price,change,i.getCropName().split("/")[1],time))
            elif k==5:
                month6.append((price,change,i.getCropName().split("/")[1],time))
            k+=1
    month1.sort()
    month2.sort()
    month3.sort()
    month4.sort()
    month5.sort()
    month6.sort()
    crop_month_wise=[]
    crop_month_wise.append([month1[0][3],month1[len(month1)-
1][2],month1[len(month1)-1][0],month1[len(month1)-
1][1],month1[0][2],month1[0][0],month1[0][1]])
    crop_month_wise.append([month2[0][3],month2[len(month2)-
1][2],month2[len(month2)-1][0],month2[len(month2)-
```

```
1][1],month2[0][2],month2[0][0],month2[0][1]])
    crop_month_wise.append([month3[0][3],month3[len(month3)-
1][2],month3[len(month3)-1][0],month3[len(month3)-
1][1],month3[0][2],month3[0][0],month3[0][1]])
    crop_month_wise.append([month4[0][3],month4[len(month4)-
1][2],month4[len(month4)-1][0],month4[len(month4)-
1][1],month4[0][2],month4[0][0],month4[0][1]])
    crop_month_wise.append([month5[0][3],month5[len(month5)-
1][2],month5[len(month5)-1][0],month5[len(month5)-
1][1],month5[0][2],month5[0][0],month5[0][1]])
    crop_month_wise.append([month6[0][3],month6[len(month6)-
1][2],month6[len(month6)-1][0],month6[len(month6)-
1][1],month6[0][2],month6[0][0],month6[0][1]])

   # print(crop_month_wise)
    return crop_month_wise

def SixMonthsForecastHelper(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.split("/")[1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 7):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year,
```

```
annual_rainfall[current_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[current_month + i - 13]))
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) ,
round(change[i], 2)])

   # print("Crop_Price: ", crop_price)
    return crop_price

def CurrentMonth(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
```

```
        if name == str(i):
            commodity = i
            break
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    current_price = (base[name.capitalize()]*current_wpi)/100
    return current_price

def TwelveMonthsForecast(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year,
annual_rainfall[current_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[current_month + i - 13]))
    max_index = 0
    min_index = 0
    max_value = 0
    min_value = 9999
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
```

```
current_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        if current_predict > max_value:
            max_value = current_predict
            max_index = month_with_year.index((m, y, r))
        if current_predict < min_value:
            min_value = current_predict
            min_index = month_with_year.index((m, y, r))
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    max_month, max_year, r1 = month_with_year[max_index]
    min_month, min_year, r2 = month_with_year[min_index]
    min_value = min_value * base[name.capitalize()] / 100
    max_value = max_value * base[name.capitalize()] / 100
    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) ,
round(change[i], 2)])
   # print("forecasr", wpis)
    x = datetime(max_year,max_month,1)
    x = x.strftime("%b %y")
    max_crop = [x, round(max_value,2)]
    x = datetime(min_year, min_month, 1)
    x = x.strftime("%b %y")
    min_crop = [x, round(min_value,2)]
```

```
    return max_crop, min_crop, crop_price


def TwelveMonthPrevious(name):
    name = name.lower()
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    commodity = commodity_list[0]
    wpis = []
    crop_price = []
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month - i >= 1:
            month_with_year.append((current_month - i, current_year,
annual_rainfall[current_month - i - 1]))
        else:
            month_with_year.append((current_month - i + 12, current_year - 1,
annual_rainfall[current_month - i + 11]))

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), 2013, r])
        wpis.append(current_predict)

    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y,m,1)
```

```
            x = x.strftime("%b %y")
            crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2)])
        # print("previous ", wpis)
        new_crop_price =[]
        for i in range(len(crop_price)-1,-1,-1):
            new_crop_price.append(crop_price[i])
        return new_crop_price


if __name__ == "__main__":
    arhar = Commodity(commodity_dict["arhar"])
    commodity_list.append(arhar)
    bajra = Commodity(commodity_dict["bajra"])
    commodity_list.append(bajra)
    barley = Commodity(commodity_dict["barley"])
    commodity_list.append(barley)
    copra = Commodity(commodity_dict["copra"])
    commodity_list.append(copra)
    cotton = Commodity(commodity_dict["cotton"])
    commodity_list.append(cotton)
    sesamum = Commodity(commodity_dict["sesamum"])
    commodity_list.append(sesamum)
    gram = Commodity(commodity_dict["gram"])
    commodity_list.append(gram)
    groundnut = Commodity(commodity_dict["groundnut"])
    commodity_list.append(groundnut)
    jowar = Commodity(commodity_dict["jowar"])
    commodity_list.append(jowar)
    maize = Commodity(commodity_dict["maize"])
    commodity_list.append(maize)
    masoor = Commodity(commodity_dict["masoor"])
    commodity_list.append(masoor)
```

```
    moong = Commodity(commodity_dict["moong"])
    commodity_list.append(moong)
    niger = Commodity(commodity_dict["niger"])
    commodity_list.append(niger)
    paddy = Commodity(commodity_dict["paddy"])
    commodity_list.append(paddy)
    ragi = Commodity(commodity_dict["ragi"])
    commodity_list.append(ragi)
    rape = Commodity(commodity_dict["rape"])
    commodity_list.append(rape)
    jute = Commodity(commodity_dict["jute"])
    commodity_list.append(jute)
    safflower = Commodity(commodity_dict["safflower"])
    commodity_list.append(safflower)
    soyabean = Commodity(commodity_dict["soyabean"])
    commodity_list.append(soyabean)
    sugarcane = Commodity(commodity_dict["sugarcane"])
    commodity_list.append(sugarcane)
    sunflower = Commodity(commodity_dict["sunflower"])
    commodity_list.append(sunflower)
    urad = Commodity(commodity_dict["urad"])
    commodity_list.append(urad)
    wheat = Commodity(commodity_dict["wheat"])
    commodity_list.append(wheat)

    app.run()
```

CROPS.PY

```
def crop(crop_name):
    crop_data = {
    "wheat":["/static/images/wheat.jpg", "U.P., Punjab, Haryana, Rajasthan, M.P.,
```

bihar", "rabi","Sri Lanka, United Arab Emirates, Taiwan"],

"paddy":["/static/images/paddy.jpg", "W.B., U.P., Andhra Pradesh, Punjab, T.N.", "kharif","Bangladesh, Saudi Arabia, Iran"],

"barley":["/static/images/barley.jpg", "Rajasthan, Uttar Pradesh, Madhya Pradesh, Haryana, Punjab", "rabi","Oman, UK, Qatar, USA"],

"maize":["/static/images/maize.jpg", "Karnataka, Andhra Pradesh, Tamil Nadu, Rajasthan, Maharashtra", "kharif", "Hong Kong, United Arab Emirates, France"],

"bajra":["/static/images/bajra.jpg", "Rajasthan, Maharashtra, Haryana, Uttar Pradesh and Gujarat", "kharif", "Oman, Saudi Arabia, Israel, Japan"],

"copra":["/static/images/copra.jpg", "Kerala, Tamil Nadu, Karnataka, Andhra Pradesh, Orissa, West Bengal","rabi", "Veitnam, Bangladesh, Iran, Malaysia"],

"cotton":["/static/images/cotton.jpg", "Punjab, Haryana, Maharashtra, Tamil Nadu, Madhya Pradesh, Gujarat", " China, Bangladesh, Egypt"],

"masoor":["/static/images/masoor.jpg", "Uttar Pradesh, Madhya Pradesh, Bihar, West Bengal, Rajasthan", "rabi", "Pakistan, Cyprus,United Arab Emirates"],

"gram":["/static/images/gram.jpg", "Madhya Pradesh, Maharashtra, Rajasthan, Uttar Pradesh, Andhra Pradesh & Karnataka", "rabi", "Veitnam, Spain, Myanmar"],

"groundnut":["/static/images/groundnut.jpg", "Andhra Pradesh, Gujarat, Tamil Nadu, Karnataka, and Maharashtra", "kharif", "Indonesia, Jordan, Iraq"],

"arhar":["/static/images/arhar.jpg", "Maharashtra, Karnataka, Madhya Pradesh and Andhra Pradesh", "kharif", "United Arab Emirates, USA, Chicago"],

"sesamum":["/static/images/sesamum.jpg", "Maharashtra, Rajasthan, West Bengal, Andhra Pradesh, Gujarat", "rabi", "Iraq, South Africa, USA, Netherlands"],

"jowar":["/static/images/jowar.jpg", "Maharashtra, Karnataka, Andhra Pradesh, Madhya Pradesh, Gujarat", "kharif", "Torronto, Sydney, New York"],

"moong":["/static/images/moong.jpg", "Rajasthan, Maharashtra, Andhra Pradesh", "rabi", "Qatar, United States, Canada"],

"niger":["/static/images/niger.jpg", "Andha Pradesh, Assam, Chattisgarh, Gujarat, Jharkhand", "kharif", "United States of American,Argenyina, Belgium"],

"rape":["/static/images/rape.jpg", "Rajasthan, Uttar Pradesh, Haryana, Madhya Pradesh, and Gujarat", "rabi", "Veitnam, Malaysia, Taiwan"],

"jute":["/static/images/jute.jpg", " West Bengal , Assam , Orissa , Bihar , Uttar

Pradesh", "kharif", "JOrdan, United Arab Emirates, Taiwan"],

"safflower":["/static/images/safflower.jpg", "Maharashtra, Karnataka, Andhra Pradesh, Madhya Pradesh, Orissa", "kharif", " Philippines, Taiwan, Portugal"],

"soyabean":["/static/images/soyabean.jpg", "Madhya Pradesh, Maharashtra, Rajasthan, Madhya Pradesh and Maharashtra", "kharif", "Spain, Thailand, Singapore"],

"urad":["/static/images/urad.jpg", "Andhra Pradesh, Maharashtra, Madhya Pradesh, Tamil Nadu", "rabi", "United States, Canada, United Arab Emirates"],

"ragi":["/static/images/ragi.jpg", "Maharashtra, Tamil Nadu and Uttarakhand", "kharif", "United Arab Emirates, New Zealand, Bahrain"],

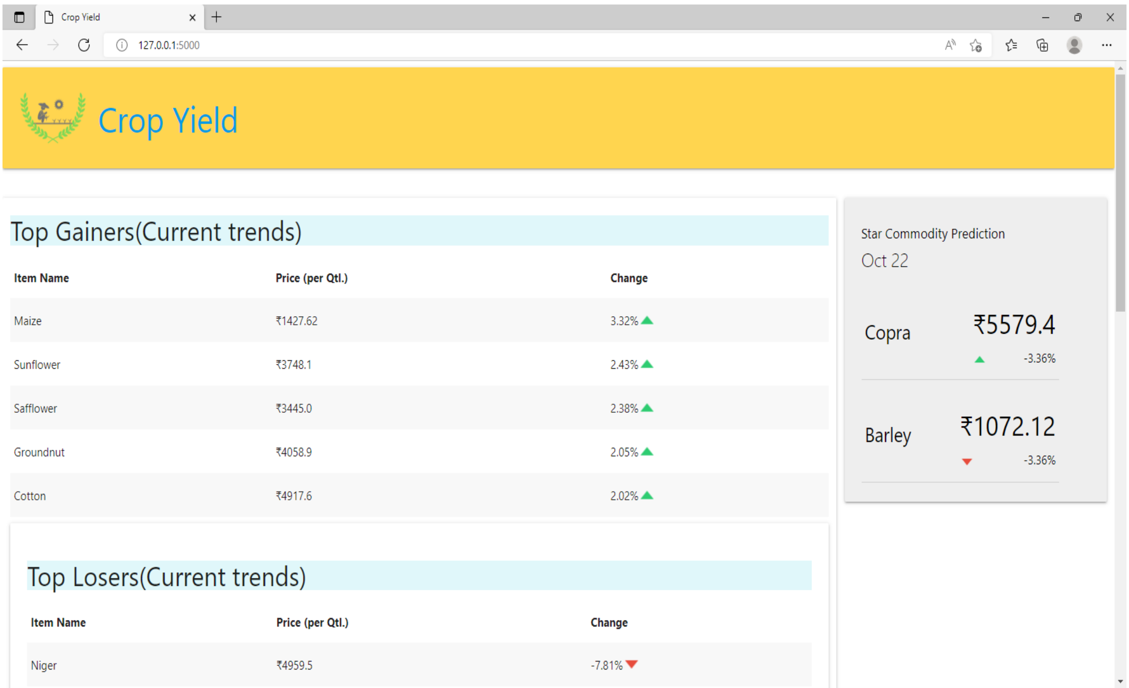
"sunflower":["sunflower.jpg", "Karnataka, Andhra Pradesh, Maharashtra, Bihar, Orissa", "rabi", "Phillippines, United States, Bangladesh"],

"sugarcane":["sugarcane.jpg","Uttar Pradesh, Maharashtra, Tamil Nadu, Karnataka, Andhra Pradesh" , "kharif", "Kenya, United Arab Emirates, United Kingdom"]

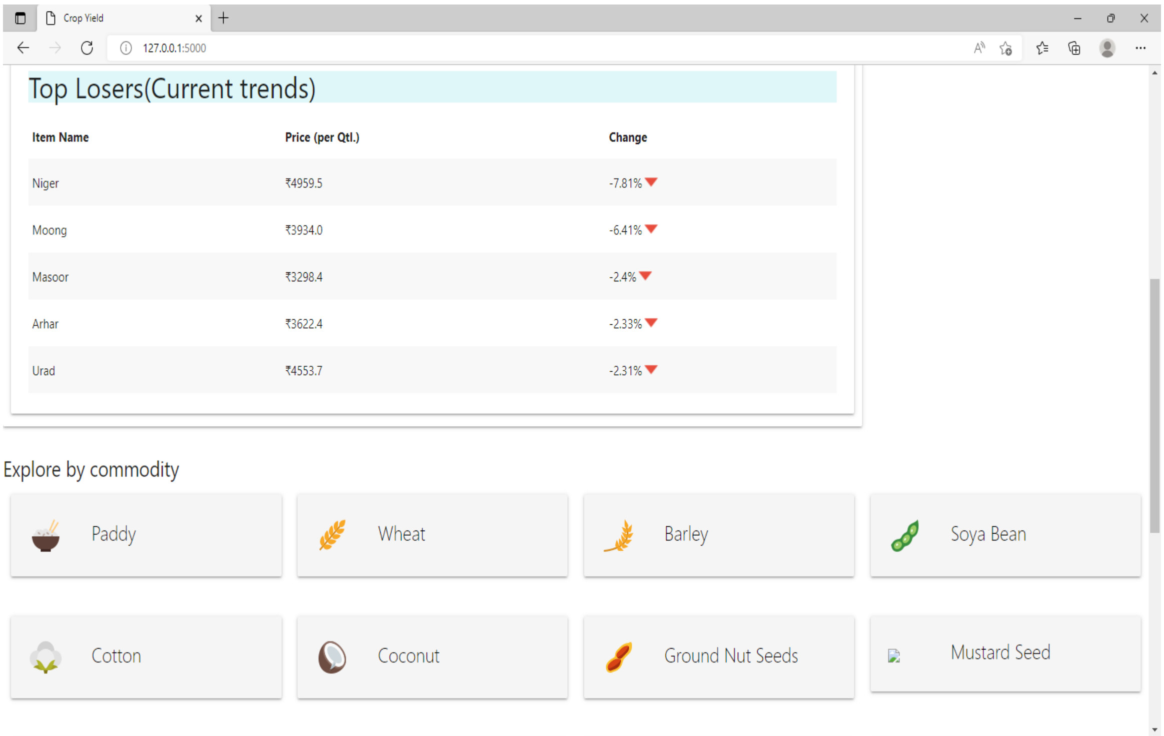
}

return crop_data[crop_name]

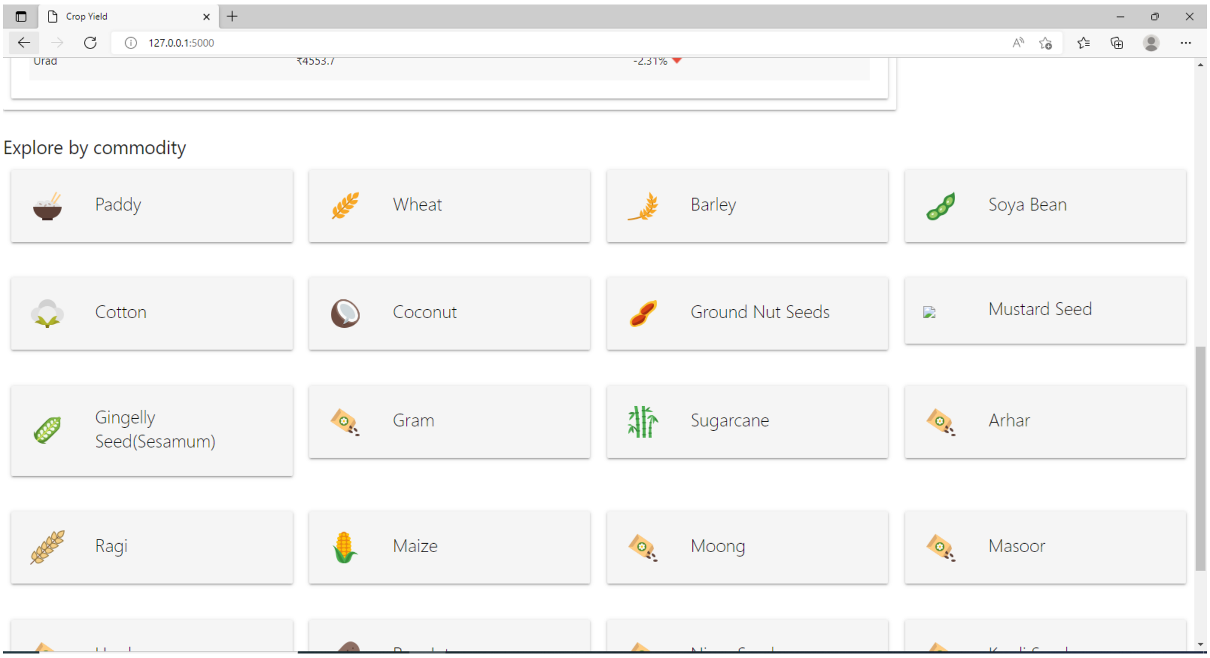
B.SCREENSHOTS



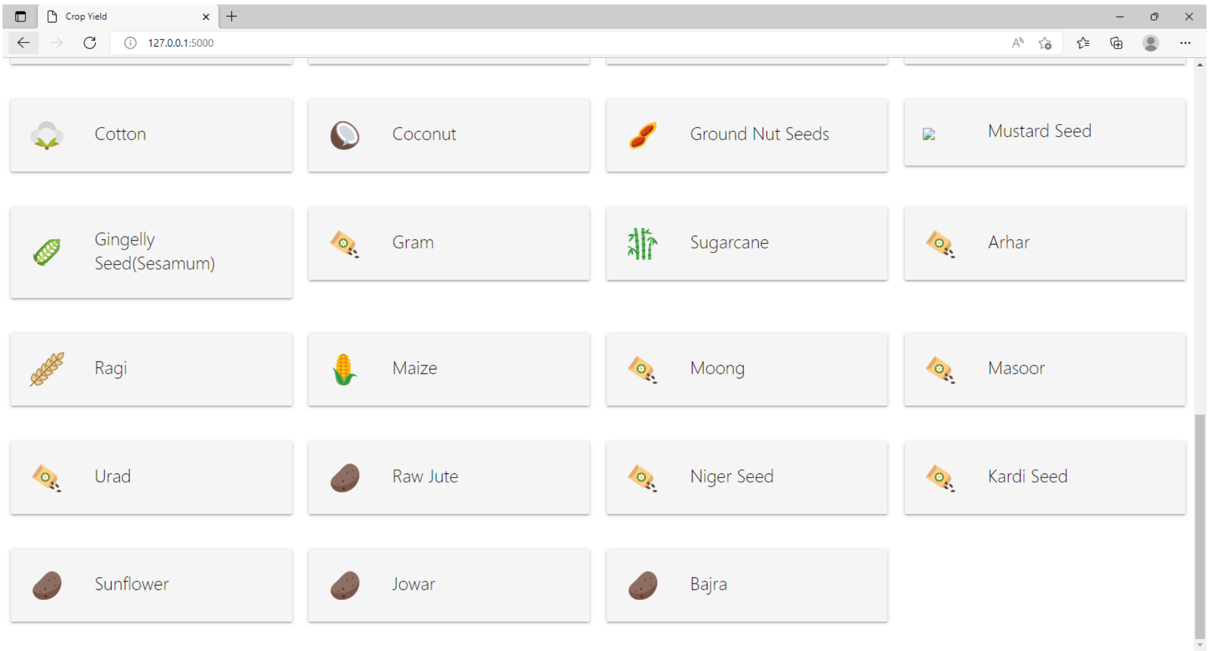
a) Star commodity prediction



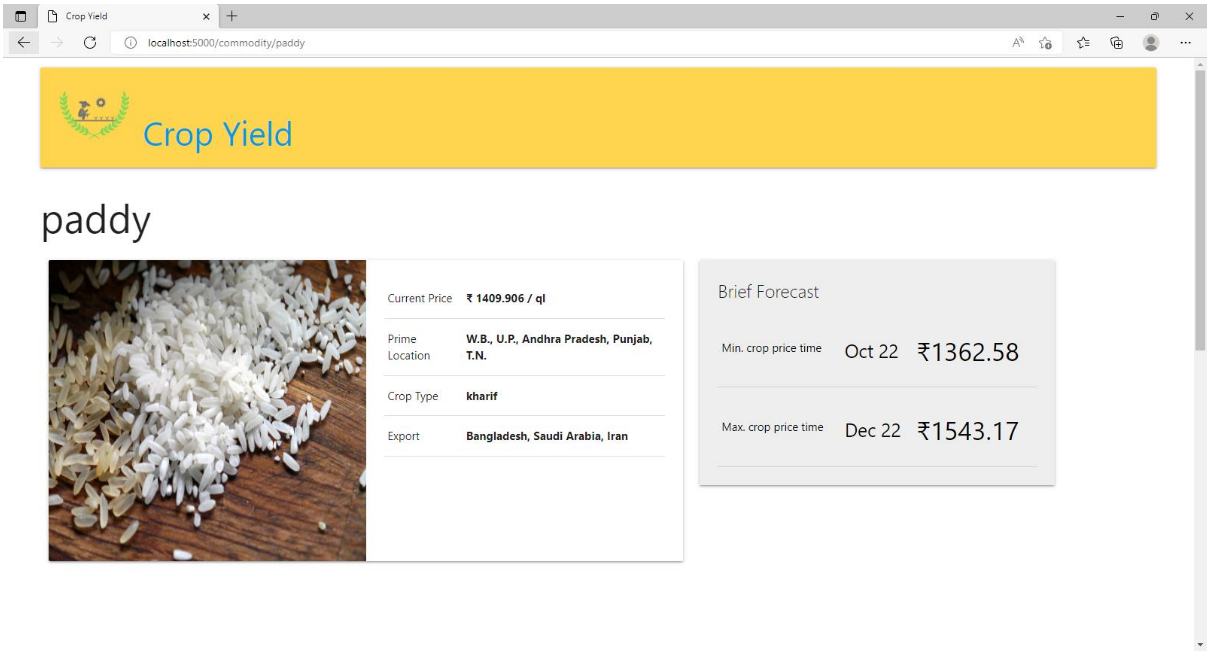
b)Explore by commodity



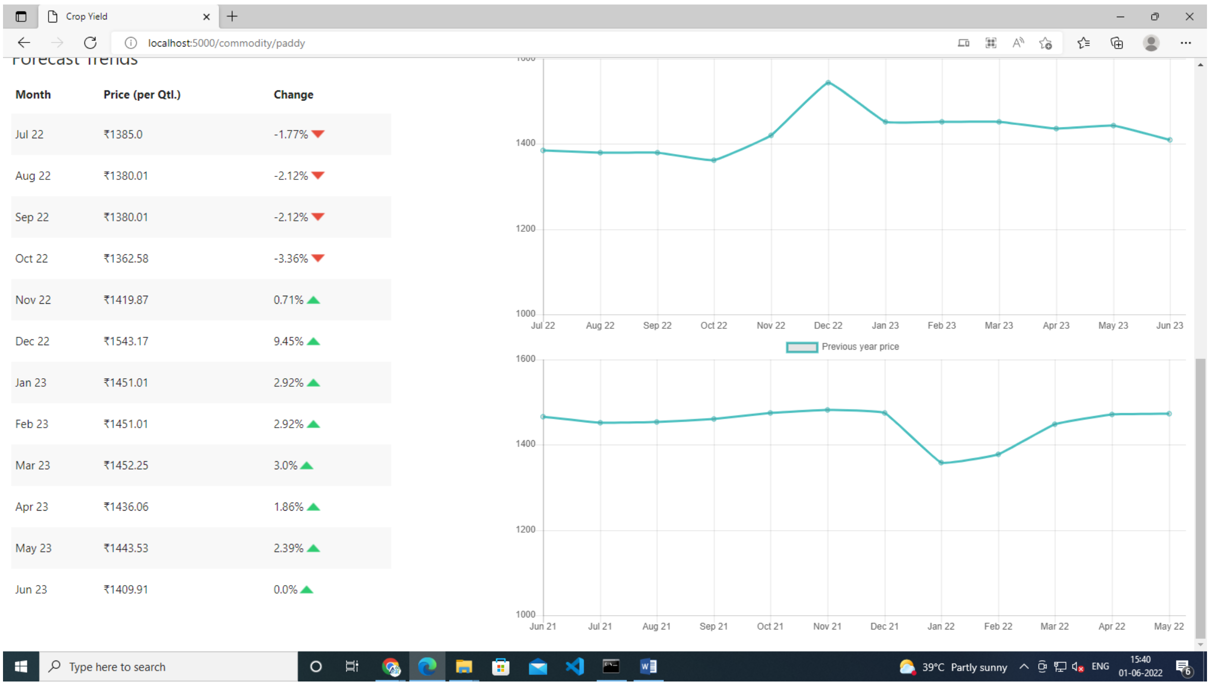
c) Explore by commodity



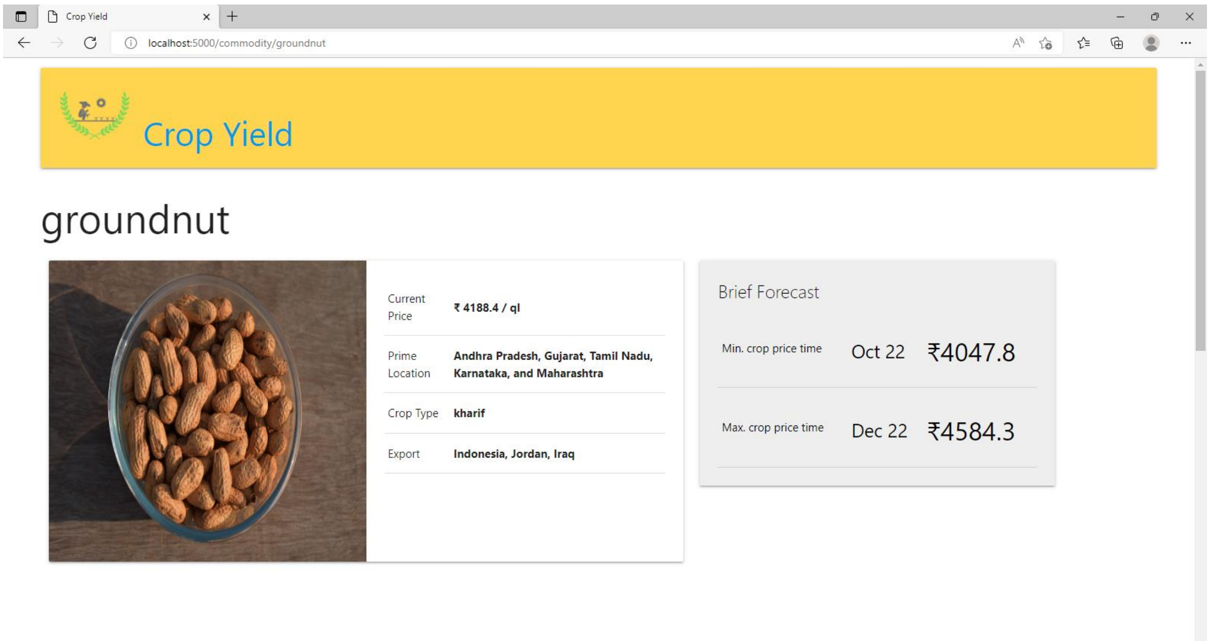
d) variaties of crops



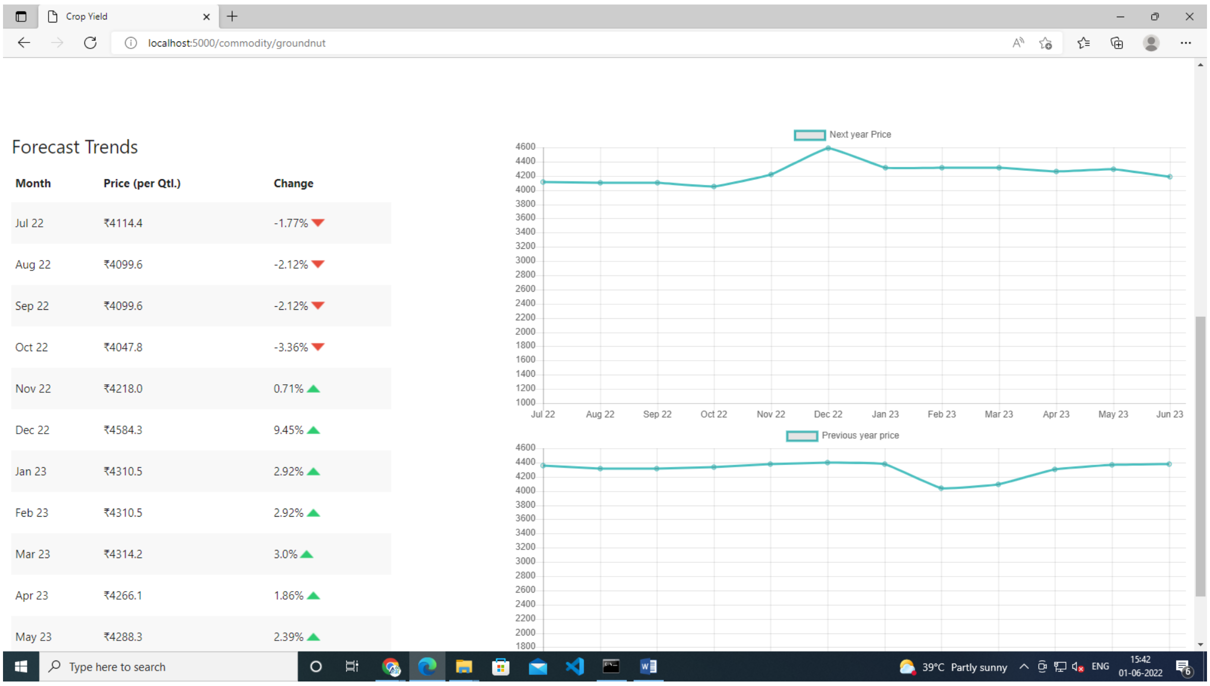
e) Paddy Crop Yield



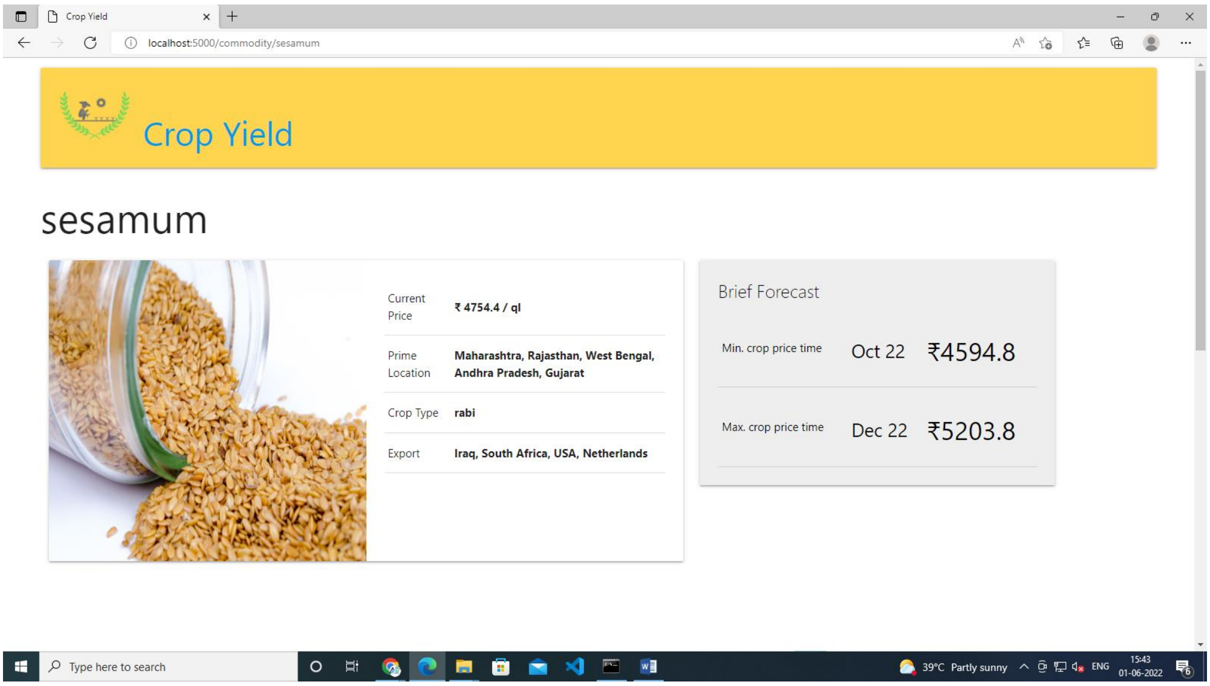
f) Paddy crop yield prediction



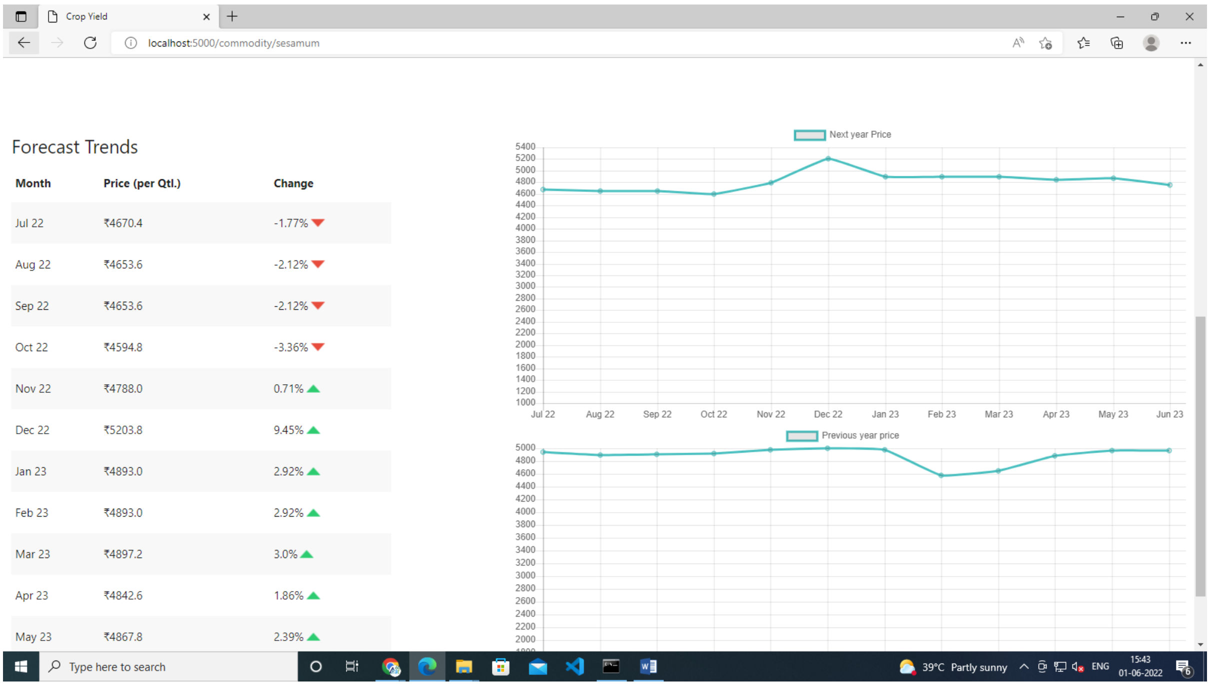
g) Groundnut Crop yield



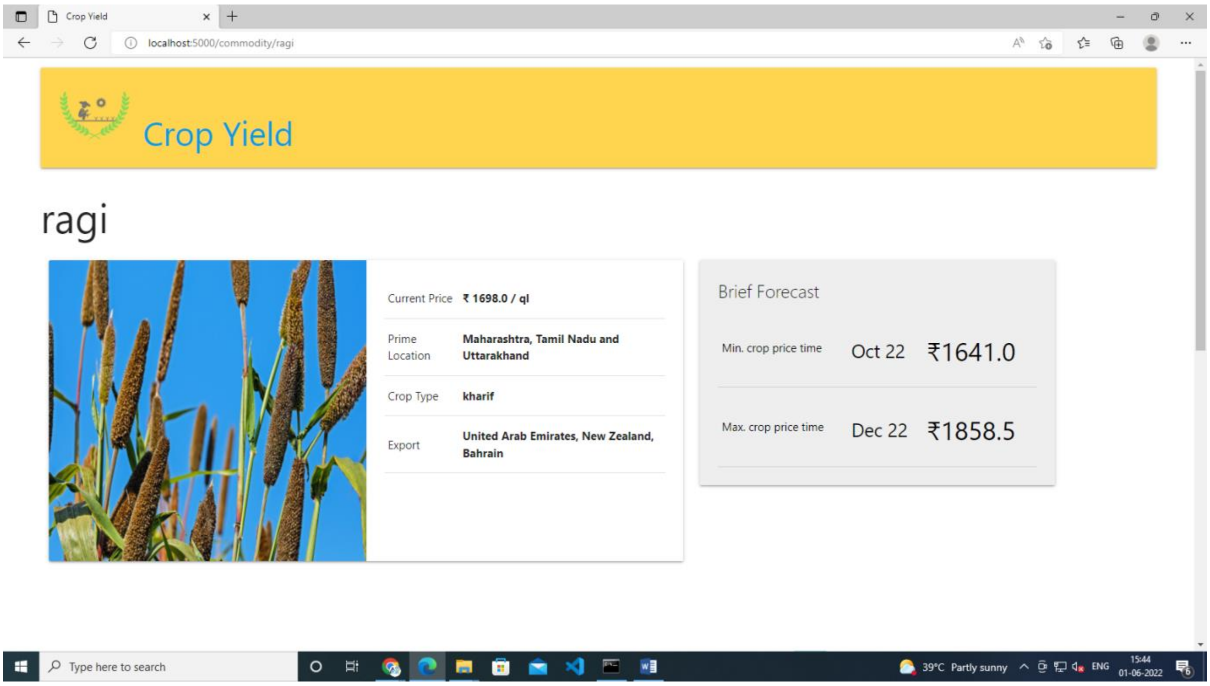
h) Groundnut Crop Yield Prediction



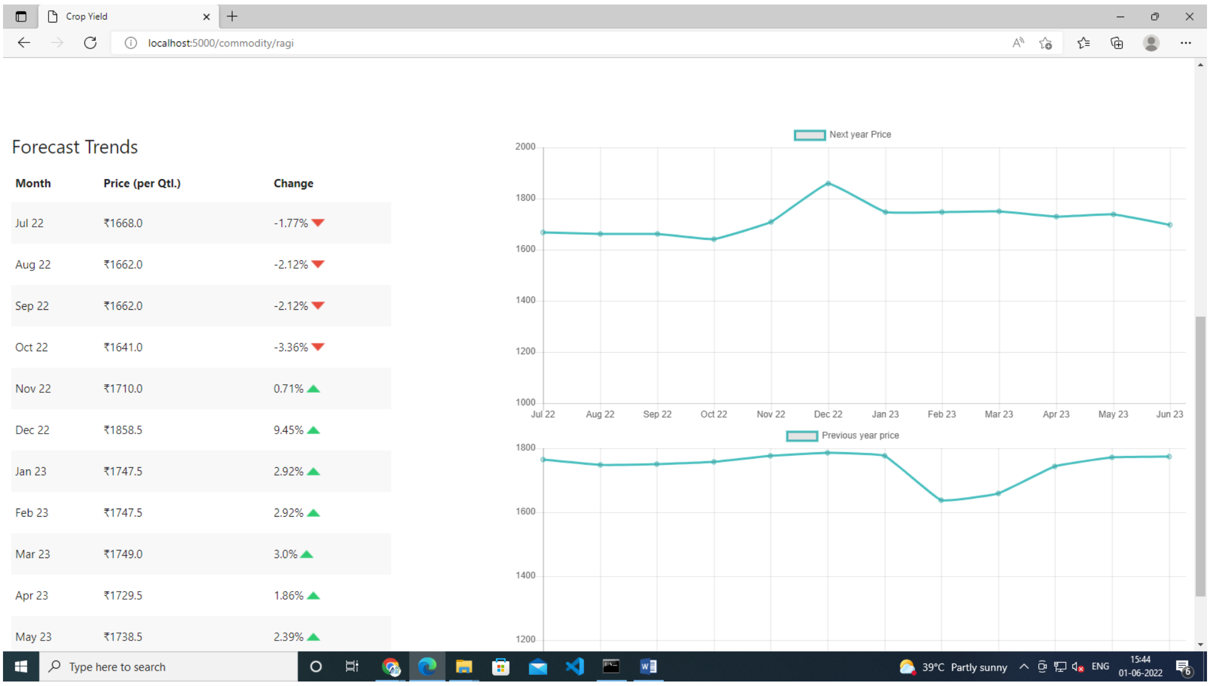
i) Sesamum Crop Yield



j) Sesamum Crop Yiled Prediction



k)Ragi Crop Yield



L) Ragi Crop Yield Prediction

C.RESEARCH PAPER

CROP YIELD ESTIMATION IN INDIA USING MACHINE LEARNING ALGORITHMS

Katta pranay
Department of Computer Science (B.E.,)
Sathyabama institute of science and technology
Chennai
Kattapranay8@gmail.com

Dr.A. Yovan Felix
Department of Computer Science (B.E .,)
Sathyabama institute of science and technology
Chennai
yovanfelix@sathyabama.ac.in

Kovvuri Sai Srikanth Reddy
Department of Computer Science (B.E.,)
Sathyabama institute of science and technology
Chennai
Saisrikanthreddy.kovvuri@gmail.com

Abstract— **India's most significant source of economic support is agriculture. The development in the populace is the significant test for food security. As a result of population growth, demand rises, necessitating that farmers cultivate more on the same amount of agricultural land to increase quantity. Advances can assist ranchers with producing more with forecast of crop yield . The primary objective of the paper is to use area, yield, production, and irrigation area to estimate crop yield. To estimate the crop yield, we used the machine learning algorithms. The cross-validation methods that were utilized for testing were mean average error, root mean, mean squared error, and squared error. The Decision tree surpasses every other machine learning approach. Crop yield prediction is the process of estimating the amount of crops that will be harvested in a given season or year. It is a critical aspect of agricultural planning and management, as it allows farmers and policymakers to make informed decisions regarding crop production, resource allocation, and food security. Various factors such as weather patterns, soil quality, irrigation, pests, diseases, and fertilization affect crop yield, making it a complex task. However, advances in technology and the availability of data have enabled the development of models that can accurately predict crop yield. These models utilize various techniques, including statistical analysis, machine learning, and remote sensing, to generate predictions that can help farmers optimize their operations and improve their yields.**

I .INTRODUCTION

Significant economic support is provided by crop agriculture, which also serves as India's primary food supply.. Numerous scientific techniques have been incorporated into agriculture to keep a healthy balance between the supply of food and the demand for it. Due to the enormous climate change, farmers struggle to make decisions that are dynamic and sustainable. Agriculture may be improved with the use of machine learning algorithms, which are useful for growing land. Therefore, crop production estimate is crucial in identifying problems with food security.

Ranchers can use crop yield evaluation to help them increase production in tolerable and good conditions and decrease production deficits in unfavourable conditions. Pesticides, fertilisers, market prices, crop yield predictions, practises, and decisions all have an effect. Using analytical data from the last year, crop harvests may be estimated using rainfall, climate, and place-wise production.

A wide range of industries, including agriculture, have benefited from the development of machine learning in recent years. Several machine learning techniques, such as support vector machines, decision trees, and artificial neural networks, have been used to forecast the agricultural production. Deep Learning , Neural Network, and Neural Network This study uses data from 1950 to 2018 to estimate India's crop yield. The expectation designed for five harvests which Maize, Jowar, Bajra, Wheat, Tobacco, and Rice. utilizing boundaries includes the area used for creation, harvesting,planting, yield, and the area covered by a water system. The decision tree and random forest are used to make the prediction.. This project aims to explore the use of machine learning algorithms to predict crop yields for various crops, including wheat, maize, and rice, among others.The project will utilize various machine learning techniques, such as regression models, clustering, and decision trees, to predict crop yields. The data used for this project will be sourced from various agricultural datasets, including weather patterns, soil quality, and historical crop yield data. Policymakers can also use the predictions to make informed decisions regarding food security, import/export, and pricing policies. Overall, this project has the potential to revolutionize crop yield prediction and enhance global food security.

II.LITERATURE SURVEY

Crop harvest may be evaluated using AI implementation approachesIn[1]Using the information, which includes total area under agriculture, irrigation water sources (tanks and wells), canal length, and average max temperature—crop output was projected.The results of a different investigation showed that the constructed computer model beat the Lasso, Shallow Neural Network, Regression tree, and DNNetwork design techniques.

In[2] For the testing of the data with estimated weather data, the RMsquare error is 50% of the standard deviation and 12% of the average yield. Conducted research with the following four goals in mind: first, examine the ability of the ANN model to estimate yields of rice and soybeans in unfavorable climates; second, assess the model's prediction ability at the state, regional, and local levels Third, assess the The objective of this investigation was to implement artificial neural networks for forecasting rice output in several areas of Maharashtra,India.

In[3] The domain of Soft Computing, Support Vector Machines (SVMs) have acquired considerable significance. These are widely used in making predictions, owing to their ability of generalization.

In[4]Method for Choosing Crops to Increase Crop Yield Planning for agriculture is important for a country with an agricultural economy and for ensuring food security. Crop selection is a crucial problem for agriculture.

In[5]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 throug hgrowingtheoptimum crop. This paper focuses on predicting the yield ofthecrop by applying various machine learning techniques. The outcomthese techniquesis compared on the b asis ofmeanabsoluteerror. 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 research makes use of Machine Learning algorithms like ANN, S.V.M, K-Means Neighbor, and Bagging algorithm to estimate crop harvest with greater precision. 745 instances of the data were used in the research. Seventy percent of the data were chosen at random to be used for model training, and the remaining 30% were utilised to test and assess the final model's performance.

III.PROPOSED SYSTEM

The implementation of the proposed system would assist the farmers in choosing the right crop that produces a higher yield and in improving our nation's agricultural practises. Additionally, it may be applied to lessen farmer losses and boost crop yields to increase agricultural capital. Thus the proposed system will help to reduce the difficulties faced by the farmers and stop them from attempting suicides and also will act as a medium to supply the farmers efficient information required to urge high yield, thus maximize profits which successively will reduce the suicide rates and lessen his difficultiesBy comparing the productivity of various crops, it is possible to increase the yield. This aids in maximising agricultural yield rates and also aids in the selection of the right crop for the farmer's chosen land and chosen season, thereby resolving their issues with agriculture. As a result, the suggested approach suggests a concept to forecast crop production. Before cultivating the field to increase production, the farmer will measure the crop's yield per acre.The system would be designed to estimate the crop yield for a given farm or agricultural region based on various environmental and agricultural factors such as soil moisture, temperature, precipitation, fertilizer usage, and other relevant parameters. The system would utilize both decision tree and linear regression algorithms to model and predict the crop yield based on these parameters.The decision tree algorithm would be used to identify the most important factors that affect the crop yield and determine the optimal values for these factors. The algorithm would create a tree-like model that splits the data into branches based on the most significant variables, allowing the system to make more accurate predictions.

The linear regression algorithm would then be used to model the relationships between the variables and predict the crop yield based on these relationships. This algorithm would create a linear equation that describes the relationship between the variables and uses this equation to predict the yield for a given set of input variables.

To use the system, farmers or agricultural experts would input the relevant data for their farm or agricultural region, including information about soil, climate, and agricultural practices. The system would then use the decision tree and linear regression algorithms to generate an estimate of the expected crop yield. Overall, the proposed system would provide farmers and agricultural experts with a powerful tool to estimate crop yield and optimize agricultural practices, ultimately leading to more efficient and productive farming.

SYSTEM ARCHITECTURE

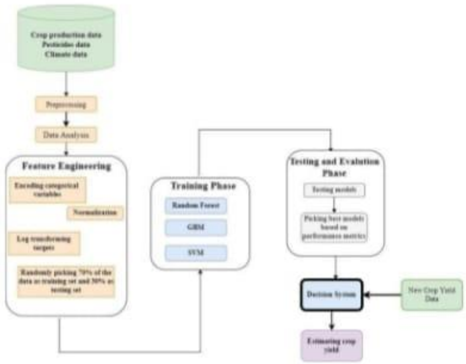


Fig.3.1 System architecture

To implement this system, the first step is to collect data about the environmental factors and the corresponding crop yields. This data will be used to train the machine learning models. The data will be split into training and testing sets, and the decision tree and linear regression models will be trained using the training set.Crop yield estimation is an important task for farmers, as it can help them make informed decisions about crop management, harvesting, and marketing. Decision tree algorithm and random forest algorithm are both popular machine learning algorithms that can be used for crop yield estimation.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

METHODOLOGY

A. *Study Region*

The authors of this study focus on India because the country's climate ranges from sultry to dry in the south to northern temp alpine. 297319 hectares of land and 179721 ha of agricultural land make up India. To play out this evaluation, six key harvests from India were picked: rice, jowar, maize, bajra, tobacco, and wheat. With 11,151,755 tonnes of quality rice exported, India is the world's leading exporter.

A.Information sources

The data for this experiment was obtained from the websites www.mdspi.gov.in and https://dataa.gov.in.,both of which are open to the public by the government. The characteristics of the obtained dataset are as follows: from 1950 to 2018, rainfall, area, irrigation area, names, seasons, output, and yield of several crops. The output of the crop over the last 68 years is shown in Figure 1.

2/6

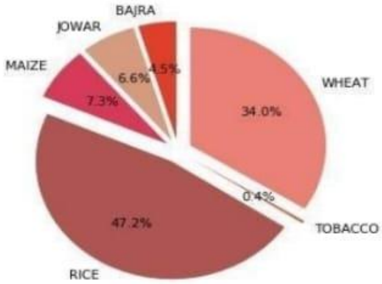


Fig. 1. Harvesting crops for 1949-2017

B. This study uses Decision Tree, Linear Regression, Lasso Regression, and Ridge Regression techniques to forecast India's agricultural productivity. Additionally used for validation are mean absolute error, mean square error, and root mean square error.

C. Used methods

1) Decision Tree

Regression and classification problems can be solved with the help of the a non-parametric machine learning approach called the decision tree. Deterministic tree algorithm performs two primary functions: first, it categorizes It first identifies the attributes required for each choice and, second, chooses the best alternative based on those traits. The plausible option is given a probability distribution by the Decision Tree algorithm. A decision tree's nodes each stand for a hallmark. Every branch leads to a choice, and the conclusion is represented by the leaf node. To begin the development of a decision tree, select a single characteristic as the root node; further data splitting is necessary to finish the tree.

2)Linear Regression

Regression and classification problems can be solved with the help of the decision tree. The decision tree algorithm serves two purposes: first, it categorises the information required for each choice, and second, it chooses which option to choose based on those attributes. The plausible choice is given a probability distribution by the Decision Tree algorithm. In a decision tree, each node stands for a feature, each branch represents a choice, and the leaf node symbolises the result. Simply choose one feature as the root node to begin developing a decision tree.

3)Lasso regression

The least Outright Shrinkage Determination Administratoris rope relapse. In lasso regression, the parameter value effects The two the size and the number of parameters are important. Increased numbers for lead the linear system toinclude additional variables.

4)Ridge Regression

Ridge regression works best when forecasting factors are highly correlated. The approach may It can be utilised to examine various regression data sets with multicollinearity. works best when a dataset has more predictor variables than observations.

Crop yield estimation is an important task in agriculture that involves predicting the amount of crop produced per unit of land. Machine learning techniques have been applied to crop yield estimation with promising results. The performance of these techniques can be evaluated using variousmetrics.

The proposed system is time-efficient because it uses machine learning algorithms to make predictions, which is much faster than manual methods. This means that farmers and agricultural businesses can quickly make decisions based on the predicted crop yields.

The methodology for crop yield prediction using machine learning algorithms typically involves the following steps:

Once the models are evaluated and selected, they can be deployed to make predictions on new data. The models can be integrated with web or mobile applications to provide real-time crop yield predictions for farmers and other stakeholders.

Overall, the methodology for crop yield prediction using machine learning algorithms involves collecting and preprocessing data, selecting important features, applying machine learning algorithms, evaluating the models, and deploying the models for real-world use. This process can help to improve the accuracy and efficiency of crop yield predictions, which can have a positive impact on agriculture and food security.

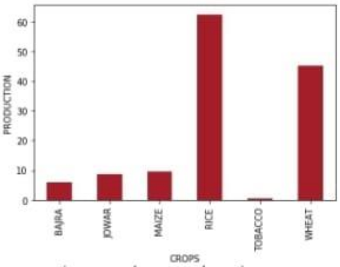
Time-series analysis involves analyzing data collected over time, such as monthly or yearly crop yield data. This method can be used to identify patterns and trends in the data, and predict future yields based on past trends. Artificial neural networks are machine learning algorithms that are designed to mimic the functioning of the human brain

.

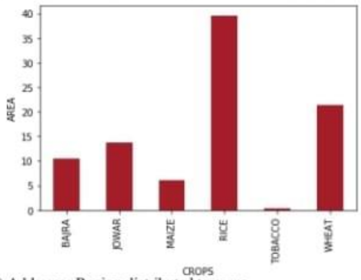
3/6

IV.Results

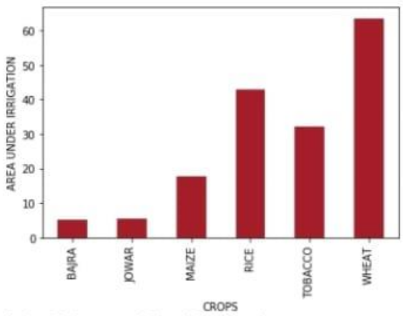
The authors thought crop was an important aspect and plotted the connection, which is seen in figure 2. The representation shows that the most rice is produced and grownin the most space. This satisfies India's position as the world's leading rice exporter. The nation produced 75% of its food grains on land dedicated to the cultivation of rice and wheat, respectively.



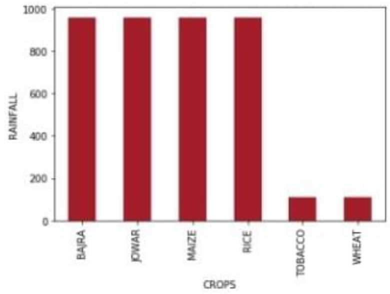
(a) Addresses connection among harvest and creation



(b) Addresses Region distributed to crops



(c) shows the relationship between yield and the subsurface water system



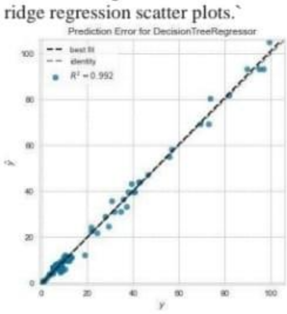
(d) precipitation and yield planning throughout the long term

Fig. 2. Connection among yield and elements, (a) Addresses connection among harvest and creation (b) Addresses Region distributed crops (c) demonstrates the region beneath the water system and the yield connection (d) precipitation and yield planning throughout the long term

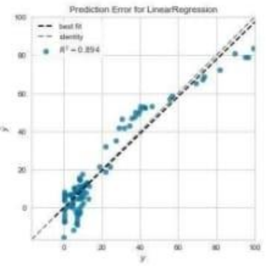
Table 1 displays the results of the predictions. According to the findings, decision trees outperform other machine learning algorithms in terms of accuracy.

Models	precision	Errors	
		MAE	RMSE
decision tree	96.73	1.56	1.71
linear regression	89.38	5.42	6.27
lasso regression	87.22	4.94	7.84
ridge regression	88.63	4.59	6.78

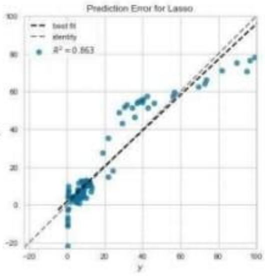
With MAE = 1.56 and RMSE = 1.71, The decision tree displays national performance. (table 1). Figure 3 predicts mistakes using a decision tree, linear regression, lasso, and ridge regression scatter plots.`



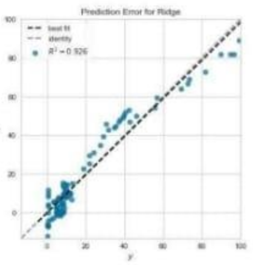
(a) decision tree



(b)linear regression



(c)Lasso Regression



(d)the Ridge regression

Fig. 3. Predicted error scatter plots to (a) thedecision tree, (b) linear regression, (c) the Lasso regression, and (d) the Ridge regression

With respective accuracy scores of 89.38 and 89.53, Linear regression and Ridge regression outperform Lasso. Utilizing statistical data, the decision tree provides the most accurate estimate of India's crop yield. Because AI does not always decode, it is a black box approach. To estimate India's agricultural production using machine learning approaches, Decision Tree outperforms the other three regression modelsin this study.

V. Conclusion

As inhabitants grows, it has become more difficult to continue the supply and demand of food chain. These past few predicting crop yield production is crucial for farmers to plan and make informed decisions regarding planting, harvesting, and marketing their crops. Machine learning techniques, such as decision trees, can help analyze large agricultural datasets and provide accurate predictions of crop yield. This can benefit farmers by allowing them to optimize their crop management practices and plan accordingly for the following year.In India, where agriculture has an important part in the economy and livelihoods of many people, utilizing machine learning methods forecasting harvest yields can be especially beneficial. By combining statistical data with remote sensing data, researchers can gain a greater comprehension of the environmental elements which impact crop growth and make predictions.

References

[1] E. J. Coyle, ―Rank order operators and the mean absolute error criterion,‖ *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 1, pp. 63–76, Jan. 1988.

[2] N. Gandhi, L. J. Armstrong, O. Petkar, and A. K. Tripathy, ―Rice crop yield prediction in India using support vector machines,‖ in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Jul. 2016, pp. 1–5.

[3] P. S. Maya Gopal and R. Bhargavi, ‒Optimum Feature Subset for Optimizing Crop Yield Prediction Using Filter and Wrapper Approaches,‖ *Applied Engineering in Agriculture*, vol. 35, no. 1, pp. 9–14, 2019.

[4] A. Gonzalez-Sanchez, J. Frausto-Solis, and W. Ojeda-Bustamante, ‒Predictive ability of machine learning methods for massive crop yield prediction,‖ *Span J Agric Res*, vol. 12, no. 2, p. 313, Apr. 2014.

[5] R. Kumar, M. P. Singh, P. Kumar, and J. P. Singh, ‒Crop Selection Method to maximize crop yield rate using machine learning technique,‖ in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Avadi,Chennai, India, May 2015, pp. 138–145.

[6]E. Khosla, R. Dharavath, and R. Priya, ―Crop yieldprediction using aggregated rainfall-based modular artificial neural networks and support vector regression,‖ *Environ Dev Sustain*, Aug. 2019.

[7] A. Mathur and G. M. Foody, ―Crop classification by support vector machine with intelligently selected training data for an operational application,‖ *International Journal of Remote Sensing*, vol. 29, no. 8, pp. 2227–2240, Apr. 2008.

[8] Kim, Nari and Lee, Yang-Won, ‒Machine Learning Approaches to Corn Yield Estimation Using Satellite Images and Climate Data: A Case of Iowa State,‖ vol. 34, no. 4, pp. 383–390, Aug. 2016.

[9] I. Nitze, U. Schulthess, and H. Asche, ‒Comparison of Machine Learning Algorithms Random forest, Artificialneural network and support vector machine to maximum likelihood for supervised crop type classification,‖ *Proc. Of The 4th GEOBIA* 35 May 2012.

[10] N. Gandhi, O. Petkar, and L. J. Armstrong, –Rice crop yield prediction using artificial neural networks,‖ in *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, Chennai, India, Jul. 2016, pp. 105–110.
[11] S. Khaki and L. Wang, —Crop Yield Prediction Using Deep Neural Networks,‖ *Front. Plant Sci.*, vol. 10, p. 621, May 2019.
[12] A. Crane-Droesch, –Machine learning methods for crop yield prediction and climate change impact assessment in agriculture,‖ *Environ. Res. Lett.*, vol. 13, no. 11, p. 114003,Oct. 2018.
[13] M. Kaul, R. L. Hill, and C. Walthall, —Artificial neural networks for corn and soybean yield prediction,‖ *Agricultural Systems*, vol. 85, no. 1, pp. 1–18, Jul. 2005.
[14] M. G. P. S. and B. R., –Performance Evaluation of Best Feature Subsets for Crop Yield Prediction Using Machine Learning Algorithms,‖ *Applied Artificial Intelligence*, vol. 33, no. 7, pp. 621–642, Jun. 2019.
[15] R. A. Schwalbert, T. Amado, G. Corassa, L. P. Pott, P. V. V. Prasad, and I. A. Ciampitti, –Satellite-based soybean yield forecast: Integrating machine learning and weather data for improving crop yield prediction in southern Brazil,‖ *Agricultural and Forest Meteorology*, vol. 284, p. 107886, Apr. 2020.
[16] –FAOSTAT,‖ 2017. http://www.fao.org/faostat/en/#rankings/countries_by_commodity_exports.
[17] Zhou Wang and A. C. Bovik, –Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures,‖ *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan.2009.
[18] T. Chai and R. R. Draxler, –Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,‖ *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.
[19] D. M. Magerman, –Statistical decision-tree models for parsing,‖ in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* -, Cambridge, Massachusetts, 1995, pp. 276–283.
[20] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*. John Wiley & Sons, 2012.
[21] M. J. A. Chan-Lau, *Lasso Regressions and Forecasting Models in Applied Stress Testing*. International Monetary Fund, 2017.
[22] S. Chatterjee and A. S. Hadi, *Regression Analysis by Example*. John Wiley & Sons, 2006.