DEVELOPMENT OF MACHINE LEARNING ALGORITHMS FOR IMPROVING CROP YIELD PREDICTION by ISMAILA OSHODI

Thesis · January 2023

DOI: 10.13140/RG.2.2.35258.57281

Ismaila Kolawole Oshodi
Bournemouth University

FACULTY OF SCIENCE & TECHNOLOGY

MSc [Data Science and Artificial Intelligence]

[2023]

DEVELOPMENT OF MACHINE LEARNING ALGORITHMS FOR IMPROVING CROP YIELD PREDICTION

by

ISMAILA OSHODI

Faculty of Science & Technology
Department of Computing and Informatics
Individual Masters Project

Abstract

Agriculture production is the economic and social backbone of the world, providing food and other products that sustain human life and support economic development. However, challenges such as increasing population growth, climate change, and industrialization can impact crop yield. Crop yield plays a crucial factor in the success of agricultural production, it's the process of estimating the number of crops that will be produced in each season or location.

Many approaches can be taken to improve crop yield, in this study, I developed machine learning algorithms to improve crop yield prediction by considering the individual factors that contribute to crop yield, including weather patterns, soil conditions, weeds and pest infestations, and decisions on what type of crops to grow. By improving the accuracy of these factors, I aim to improve crop yield predictions.

To achieve this, I developed five modules that contribute to crop yield prediction: a crop recommendation module, a weather prediction module, a fertilizer recommendation module, a plant disease identification module, and a weed detection module. These modules were integrated into a user-friendly web interface that farmers could use to input data and receive predictions based on the modules. The performance of the algorithms was measured using accuracy, and they all achieved high accuracy between 53% and 100%. This study also analyzed the impact of climate change on agriculture and climate-smart agricultural practices.

Overall, the development of machine learning algorithms for improving crop yield prediction represents a significant step forward in addressing the challenges facing agriculture and supporting the economic and social development of communities around the world. Adopting these systems would help improve crop yield predictions, help farmers make informed decisions about their farming operations, reduce GHG emissions in the agricultural sector, and support the adoption of sustainable farming practices.

Keywords: Agriculture, Crop yield, machine learning,

Dissertation Declaration

I agree that should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. Any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history, or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

Signed: ____.

Name: Ismaila Oshodi

Date: 15th January, 2023.

Programme: MSc Data Science and Artificial intelligence

Original Work Declaration

This dissertation and the project that it is based on are my work, except where stated, in accordance with University regulations.

Signed: ____.

Name: Ismaila Oshodi

Date: 15th January, 2023

Acknowledgements

To everyone who has attained to make an impact on me throughout this journey, I say jazakumullahu khairan

To all my friends and family who have supported and encouraged me, you are truly the rock of my oasis.

My sincere gratitude to my supervisor, Dr Avleen Malhi for your Guidance, and support throughout this process, I express my heartfelt appreciation.

TABLE OF CONTENTS

LIST OF FIGURES

List of Table

1 INTRODUCTION

1.1 Background of study

Agriculture has been a vital part of human society since the dawn of civilization, it plays a critical role in our society; the agriculture sector contributes immensely to the value and structure of a country's economy, it is also an important factor in the economic growth of a country, providing employment, contributes to food security, and generates foreign exchange earnings, it is considered an important pillar of the world's economy (Bhanumathi et al. 2019).

The population of the world has quadrupled in the previous century, the global population has grown from 1 billion in the 1800s to 8 billion in 2022 and it's expected to keep growing, according to the most recent UN estimates, the population of the world would be about 9.7 billion by mid-2050s (United Nations Department of Economic and Social Affairs, Population Division, 2022). Rapid population growth, depletion of agricultural resources and climate change have raised global food consumption, and demand is expected to keep increasing for decades.

Food production is threatened by several factors, including economic development, pest and disease, the effects of climate change such as severe temperatures, drought, floods, changes in rainfall patterns, extreme weather events, and an increase in the frequency and intensity of forest fires, among others. According to the United Nations Food and Agricultural Organization (FAO), food production will need to increase by about 70% in 2050 to meet the demand for food (Arsene, M.B, 2021; Food and Agriculture Organization of the United Nations (FAO, 2009)).
To help farmers satisfy the rising demand for food crops, it is imperative that novel and innovative agricultural production techniques for enhancing crop yields be developed. Farming has benefited from breakthroughs in artificial intelligence, climate-smart agriculture, and other agricultural technologies. (S. Vadlamudi, 2019)

A subset of artificial intelligence termed machine learning examines data to find patterns and make projections. Economics, manufacturing, and finance are just a few of the industries that have successfully integrated machine learning (van Klomenburg et al., 2020) for example, upstart a money lending platform for consumer loans developed a creditworthiness assessment tool to predict a potential borrower's income and risk of default; They combined credit report data with information about the borrower's work history, educational background, GPA, and scores on standardised tests to create a static model that uses artificial intelligence to assess the borrower's financial capacity and ability to repay the loan (upstart, 2022).

Machine learning has been successfully implemented into agriculture; it can be applied to all stages of agricultural production like cultivation, harvesting and post-harvesting (storage and marketing) (Meshram, V. et, 2021); machine learning can be used to make predictions of the individual factors that can improve crop yield, including crop recommendation, weather and rainfall predictions, fertilizer recommendations, and plant disease predictions; these factors will help improve the crop yield predictions.

The use of machine learning algorithms for crop yield prediction is still in its early stages of development. However, it is a rapidly growing area with the potential to significantly improve crop yields []; Crop yield prediction is based on various data that is gathered and analysed to increase crop yield; it is influenced by many indicators that are considered as inputs, such as agricultural practises, biotic factors like pests and diseases, and environmental elements like weather and soil characteristics; increasing crop yield necessitates an understanding of these indicators. (Liane T.N. and Charles M.S., 2020; Meneka K. and Yuvaraj N., 2016).

Farmers can evaluate crop yields with the use of prediction models; how much crop production they can expect, as well as how to enhance their farming practises to increase yields and profits while enhancing the quality of their output to satisfy market demand. (Dahikar and Rode 2014)

1.2 Smart agriculture

Human has been cultivating crops for over 10,000 years and agricultural technology has been evolving since, better and more efficient ways of farming have been emerging as farming has changed from using simple handheld tools to large, mechanised equipment, and farmers have begun embracing a new method of farming by adapting to high tech mechanization.

Technology advancement and innovations have improved every industry across the world, In the 21^{st} century today it has shaped the agricultural sector, Farming methods have become automated, data-oriented, flexible, more redefined and less manual, it uses merging technologies to continue to modernize and enhance traditional agriculture.

Among the challenges of climate change, agriculture must remain innovative and use e to meet the increasing demand of the human population for higher nutritional food crops.

Smart agriculture refers to the integration of information and communication technology into agricultural equipment, machinery and sensors for use in crop cultivation, food production and animal husbandry (Virk, A.L., et al. 2020) it involves using the necessary infrastructure to leverage cutting-edge technology to perform and optimise complex farming operations.

smart agriculture employs varying information and data technologies for monitoring, tracking, automating and analysing farming operations, they include drones, wireless sensor networks, software, the internet of things (IoT), big data, blockchain, Robots, Artificial intelligence and machine learning etc. they use several types of Interconnected to a network to collect, store, and analyse data of factors that determine farming outcomes (Suma N, et al 2017).

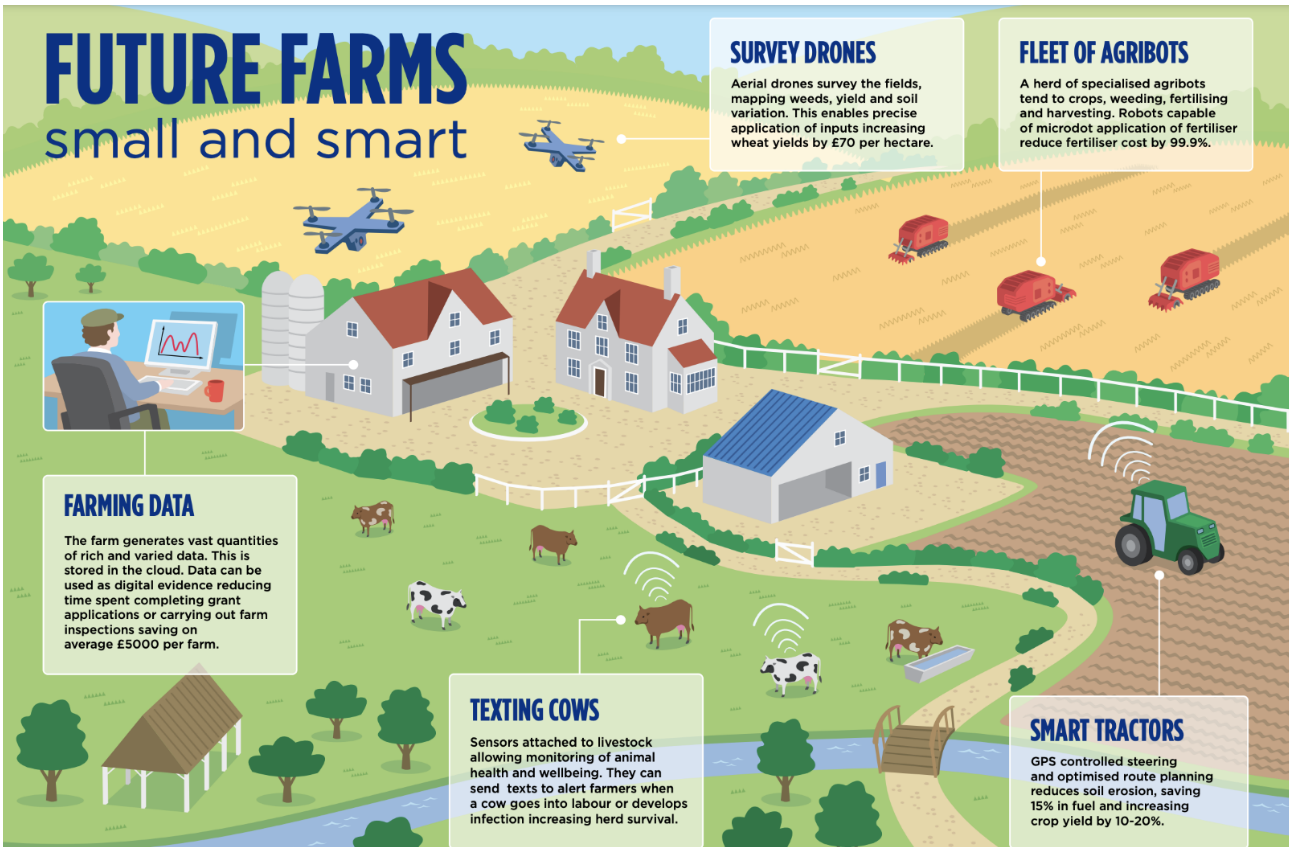

Figure 1: smart farming (Nesta, 2022)

The main goal of smart farming is to increase the predictability and efficiency of agricultural operations by collecting data and analysing it using computing technology. Famers can install smart agriculture devices in fields for crop management and use them to collect and analyse specific data and variables such as temperature, precipitations, and overall crop health, this real-time data will be used to suggest a practical plan for harvesting, propose precautions to prevent crop damage and maximising crop yield productivity. (Boursians A.D., 2022)

Smart agriculture devices can also be used in crop management for automating tasks according to the demand of the farmer such as planting and sowing, harvesting, automatic irrigations, weather forecasting, crop spraying, soil and field analysis, weed, pest and disease control and maintenance.

Combining data from numerous agricultural sensors may be used in smart agriculture to monitor climatic conditions. These sensors gather information about the environment and transfer it to the cloud, which enables farmers to adapt their reaction to the climate and effectively track crop development. (), devices for smart agriculture can also support the tracking of livestock's locations and health using a variety of sensors. These sensors will speed up the diagnosis and treatment of livestock by spotting disease and containing it, monitoring feeding, and speedy recovery of livestock that has been stolen.

With smart agriculture farmers or growers can achieve better control over production using real-time data from interconnected sensors, they can use their smartphones to remotely monitor crops, livestock, and equipment and run statistical predictions for their crops and livestock, this will enable them to refine the standard of products and achieve efficiency in farming through automation, however, they are other benefits of smart agriculture including a reduction in production cost of farming activities, removing human error in agriculture, minimise environmental constraints, improve food security, optimise crop yields and boost natural resources efficiency.

The data collected can be used to improve the business side of agriculture and analysed using machine learning to make informed decisions to improve the efficiency of their day-to-day work. Smart agriculture benefits the farmers; however, the innovative technology has its financial implications, among other challenges including technical know-how, hardware limitations, installation and maintenance, and data security among others.

Following the plant and genetics revolution (Andrée, 2005), smart agriculture can deliver a more productive and sustainable form of agricultural production using the combined application of data-driven analytics and IOT, making possible new ways of planting, storing, processing, managing and marketing agriculture and agricultural product

1.3 Problem definition

The impact of climate change on agriculture, the degradation of fertile farmland, and the rising human population have led to growing demands for higher nutritional foods which affect the balance between supply and demand for food crops (Chakraborty S.K., 2022).

Crop yield prediction is an important agricultural and economic problem. Farmers, researchers, policymakers, and agricultural investors have always been faced with the problem of not knowing how to improve their yield, how much harvest they will produce and how the agricultural sector will perform in a particular season.

Accurate agricultural yield forecasting has numerous benefits. For agricultural planning, policymakers require precise and accurate forecasts; these forecasts are used to decide on import and export policies as well as agricultural support and subsidy programmes, agricultural investors need crop yield prediction for financial evaluations when making investment decisions.

For a variety of reasons, farmers rely most on projections of crop yield. It can help farmers decide what to grow and when to sow it using high-quality data and machine learning models, this will enable them to maximise their resources, increasing production and profit while meeting the population's higher nutritional needs and overcoming different threats to global food security by ensuring theirs enough food to meet demand. Additionally, crop yield prediction will also allow farmers to make reliable and data-driven financial planning decisions.

Crop production is a complex phenomenon that is influenced by a variety of factors, including agricultural techniques or technology employed, as well as environmental and biological factors. Weather, climate, soil fertility, and water are examples of environmental variables, whereas crop genotype, pests, and plant diseases are examples of biological factors. These factors account for yield disparities between farms and regions. (khaki s. and Lizhi W. 2019),

Before the emergence of machine learning technology in agriculture, a farmer faced various challenges; before cultivation, He made decisions about what, when, and how to cultivate a crop based on insufficient and inaccurate information; he attempts to maximise the soil nutrient but is unaware of which crop will be best suited to accomplish that, he is unaware of the environmental conditions for the planting season or how the biological factors like crop genotype impact his yield.

During cultivation, he must cope with the issues of inadequate or excessive rainfall, pests, and plant diseases, he forecasts his crop yields by observing how his developing crop is responding to the circumstances of the environment, based on his knowledge of the field and crop. (shar v. and Prachi p., 2018), this procedure generates false results. The way he overcomes these challenges before and throughout the planting seasons will determine his crop yield, necessitating the development of novel and integrated approaches to improve crop yields.

Technologies based on machine learning provide innovative and comprehensive approaches to these challenges, they offer solutions to the problem of improving crop yields and other agricultural challenges a farmer may face.

The crop yield is predicted using machine learning to provide an estimate of the crop output. (Islam et al., 2021). Today, with the development of the Internet of things (IoT), smart agriculture and data analytics, agricultural activities can generate hundreds of data per day; these data can be used to predict crop yields by incorporating historical data from the past with other relevant variables such as agricultural practises, environmental factors and biological factors into a machine learning algorithm in order develop a model; data of these interrelated factors must be of high quality, with the correct information, and typically varying from field to field, to build an accurate machine learning model.

An accurate machine learning model will surely have a great impact on agriculture, Policy makers, farmers, investors and consumers would benefit from it immensely. Policymakers would be able to make decisions regarding subsidies and import and export decisions, farmers would improve their crop yield, and select the best crops to cultivate, crop yield prediction can assist in the identification of plant disease and pest management, automate simple and complex farm activity, improve the productivity and efficiency of farming, and make informed data-backed management and financial decisions,

1.4 Aims and objectives

The research aims to use publicly accessible datasets to create a prediction model for improving crop yield.
The objectives of this research include:

- Objective 1: Present an overview of machine learning applications in agriculture
  a) Review the applications of AI and machine learning in agriculture, as well as their benefits and drawbacks.
  b) Examine the most recent studies on enhancing agricultural yield prediction. Evaluate smart agriculture practices,
  c) Conduct a review of climate change on agriculture, climate-smart agriculture practices and smart agriculture

- Objective 2: Implement machine learning techniques to predict crop yield
  a) Select and pre-process relevant data from publicly accessible datasets.

b) Train and evaluate machine learning models for crop yield prediction.
c) Compare the performance of different machine learning approaches and select the best model.

- Objective 3: Develop a web-based interface for the crop yield prediction model
  a) Design and implement a user-friendly web interface for farmers to input their data and receive personalized predictions.
  b) Utilize methods to assess the effectiveness of the prediction model for all crops.
  c) Determine the most appropriate machine learning approach for all crops to enhance yield prediction accuracy.

1.5 Research questions

The following are the research questions for this project:

a) How does climate change impact agriculture, and what are the risks and adoption levels of climate-smart agriculture practices?
b) What are the limitations and challenges faced by farmers in implementing AI in agriculture?
c) How can machine learning techniques be used to improve crop yield prediction, and what attributes have been utilized in previous research in this area?
d) How can AI be utilized to enhance crop yield, and what are the potential challenges in implementing these technologies in agricultural practices?

1.6 Project structure

This project is structured into five chapters, the organisation is as follows; Chapter 1 the introductory chapter provides a brief description of the project, its background, and problem definition, it also includes a brief review of smart agriculture, the aim and objectives of the project and the research questions of the project.
Chapter 2 focuses on the literature review, where related work on improving crop yield was reviewed, it also contains the review of the impact of climate change on agriculture, climate-smart agricultural practises, and the applications of Artificial intelligence and machine learning in agriculture with their advantages and limitations.
Chapter 3, the methodology consists of the proposed system, techniques and approaches used. Data acquisition and pre-processing were also analysed here. Chapter 4, experimental setup, is where the proposed system was implemented, the result was analysed and a web a based interface was developed. Chapter 5, Evaluation, and conclusions, here critical evaluation of the aims and objectives was carried out also recommendations for future work were done.

2 LITERATURE REVIEW

2.1 AI and machine learning in agriculture

Artificial intelligence (AI) is a type of intelligence shown by machines. It has helped change the way we live and solve problems, creating new opportunities for economic growth. AI has been used to find solutions to problems in various fields, including agriculture, business, education, finance, healthcare, and retail.

AI has been used to improve the agriculture industry by making it more efficient and effective. It offers innovative solutions for modernizing the sector. Although the agriculture industry has quickly adopted AI, it is still a new field being used in many areas of agriculture, such as crop management and livestock management. (Farooq M.S et al., 2020; Dharmaraj V., et al., 2018).

2.2 Crop management

Crop management is a group of agricultural practices performed to enhance the growth, development, and production of crops. The adoption of best crop management practices has been known to improve crop productivity; weed control, crop disease diagnosis, pest management, irrigation, yield prediction, and other crop management procedures have all been improved with the use of artificial intelligence techniques and machine learning algorithms.

2.2.1 Crop disease detection

All species of plants are subject to disease. crop disease is a process where an abiotic or biotic entity interferes with normal functions and hinders the growth of a plant. Abiotic factors are caused by conditions external to the plant, these include environmental conditions (temperature, and humidity) and soil conditions (PH level and moisture level of the soil); Biotic entities are usually caused by living organisms, known as plant pathogens, examples include bacteria, viruses, fungi, and nematodes etc. Biotic entities are infectious, and a farm can see a rapid spread of infectious biotic disease easily (smith D.R. & White D.G., 1988).

The occurrence of crop diseases varies seasonally, depending on the prevalence of a pathogen, conditions of the environment, the varieties of crop grown, blackspot, blight, yellowing of the leaves, growth retardation, rot, spots on leaves, and other physical characteristics of the plant are used to identify plant diseases, crop disease poses a serious challenge to the production of food and food security, as it lowers the quality and the number of yields.

Identification of plant disease is a great challenge, traditionally they are several ways to detect plant disease, however, these methods are ineffective, and time-consuming as each plant must be evaluated individually, however, with the help of machine learning combined with IoT and sensors plant disease can be detected and managed effectively (Kumar M. S., 2022).

The focus of recent research on plant disease has been on image processing-based machine learning models; In 2020, Ashok S. et al. propose to diagnose the Tomato Plant Leaf disease using image-processing techniques based on picture segmentation, clustering, and open-source algorithms. The suggested method uses a CNN algorithm for hierarchical feature extraction to map the pixel intensities of the input image and compare them to those of the image from the trained dataset. All adjustable leaf section parameters are optimised by reducing the error across the training set. The comparison image is then classified as either having a diseased leaf or a normal leaf as part of the deployed and implemented image classifier approach. The study's accuracy rate was 98 per cent

Using straightforward photos of healthy or sick leaves, Ferentinos K. P. created a convolutional neural network in 2018 to recognise and diagnose plant illnesses. The models, which are based on convolutional neural network architectures, were trained using a publicly available dataset of 87,848 images captured in both laboratory settings and actual agricultural areas. On 17,548 previously undiscovered images, the best model architecture, a VGG convolutional neural network, achieved a success rate of 99.53 per cent (top-1 error of 0.47 per cent).

2.2.2 Weed detection.

Weeds are undesired plants that emerge naturally alongside crops and compete with desirable plants for light, growth space, water, and nutrients. They also spread disease and draw harmful insects and pests (Liu bo and Bruch R, 2020). Weed control is a critical tactic for increasing crop production because weeds are one of the major factors affecting agricultural output and harm crop yields and quality. Different operations, such as hand cultivation using homemade equipment, cultural management (using crops that can destroy the existence of weeds), and chemical bombardment with herbicides have all been tried to control weeds.

Literature has suggested variable spraying techniques to avoid waste and herbicide residue issues brought on by full-coverage spraying. Weed identification using machine learning offers a way to reduce or eliminate the usage of herbicides (Wu, z et al, 2021). To identify crops and weeds in the field in real time and provide useful sensing information for site-specific weed control, several researchers have investigated computer vision mixed with image processing methods and deep learning (Wang A et al., 2019).

Islam, N. et al. 2021 investigated how machine learning and image processing techniques might be used to categorise crops and weeds from UAV photos. Images from an Australian chilli crop field were obtained and pre-processed utilising image processing methods that enable features to be extracted to discern between crop and weed characteristics. Despite using distinct methods, RF and SVM were able to recognise weeds from RGB pictures with 96 and 94 per cent accuracy, respectively.

2.2.3 Pest management

Pest in farms refers to insects and invasive animals that attack and eat food crops and livestock for nourishment while harming them, impairing the growth and yield of agricultural goods, and transmitting disease. According to Almeyda E. et al. (2020), this is one of the biggest difficulties with crop management.

Pest management involves using chemical, biological, cultural, physical, and other strategies to bring the number of pests population to an acceptable threshold. The identification and monitoring of pest and insect populations are advantageous to growers and must become a top priority to create an ideal and prime harvest.

Farmers typically employ a variety of techniques to manage the insect population and determine the extent of the damage, including cultural techniques, the use of plants that act as insecticides and repellents, as well as periodic walks around the entire farm while carefully inspecting a few plants for pest activity, and the use of monitoring traps and pesticides. This method is ineffective and wasteful because it encourages the indiscriminate use of pesticides on farms (Abate T. et al, 2000).

Farmers can now use an integrated approach to the prevention and control of pests, to cause the least amount of disruption to the agricultural ecosystem by keeping the use of pesticides and other forms of intervention only to levels that are justifiable economically and ecologically. Pest management using artificial intelligence and machine learning involves solving pest problems while minimising risk to people and the environment. To accomplish this, sensors, intelligent monitoring systems, artificial intelligence agents, and machine learning algorithms are used to detect, monitor, and access the pest population (Durgabai R.P.L and Bhargavi p., 2018).

Artificial intelligence will help farmers in predicting pest attacks, grow a healthy crop that can withstand pest attacks, create conditions that are unfavourable to the pest, automate the detection of the pest in real-time and develop strategies to keep the pest away, that avoid reliance on pesticides.

Based on metrological data, satellite earth observation, and previous trap captures, Nanushi O. et al., 2022 found a significant cotton bollworm presence in their study. The air temperature and relative humidity were environmental factors considered by Markovic D. et al. in 2021 for their machine learning model to predict the appearance of pests. The research had an accuracy of

86.3% and a false detection rate of 11%. In another study, Skawsang S. et al., 2019 developed a predictive model for rice pests using ground-based weather factors, time series of satellite-derived vegetation indices, and machine learning techniques to create the forecast using multiple machine learning algorithms. satellite data, meteorological data, and pest data were acquired from the TERRA MODIS satellite, weather loggers at the sites, and light trap stations deployed, respectively.

2.2.4 Automated farming

Automated farming is the practice of utilising a variety of technological developments and breakthroughs to automate a variety of farming processes to optimise food production, improve the quality of crops, and decrease labour and time-intensive activities. (Jha k. et al., 2019).
To automate specific tasks in agriculture Durai S.K.S., et al. 2022 recommend employing several artificial intelligent agents, sensors, blockchain, IoT, expert systems, and machine learning algorithms to automate particular jobs in agriculture. Among the automated agricultural equipment that can be used in agriculture are robots for sowing and weeding, harvest automation devices, automatic irrigation, autonomous tractors, drones for fertiliser spraying, etc. (Krishnan A. and Swarna S., 2020)
Choudhary S. et al. proposed an autonomous crop irrigation system in 2019. In this study, an automatic irrigation system based on artificial intelligence and the internet of things was developed, this system can water fields using soil moisture data. Rainfall prediction is done before irrigation to identify when irrigation is necessary based on the weather and current soil moisture conditions. automated farming equipment might be multifunctional or specialised, for example, Dhole S. et al. 2022, built a specialised robot that sows seeds, Akhila G. & M. B. Srinivas 2011 created a multipurpose automation robot that can sow seeds, plough fields, and cover the seeds with soil.

2.2.5 Agricultural marketing

The commercial activity of getting agricultural products to consumers is known as agricultural marketing. Strategic planning, organising, and processing of agricultural products are some of the services provided in this process to minimise post-harvest losses, make the best use of available resources, and increase profit (Aravatagimath A., et al., 2021).
Artificial intelligence uses Predictive analysis to collect, analyse and monitor data to allow farmers to study the current and future supply and demand trends for both domestic and international markets, analyse the market structure and determine prices for agricultural products based on

supply and demand, make data-backed decisions for processing, advertisement, storage and distribution of the product.
A combination of other technologies, such as blockchain, cloud computing, report visualisation tools, etc., is utilised alongside artificial intelligence.

2.2.6 Crop monitoring and yield prediction

Crop monitoring uses numerous sensors, artificial intelligence, cloud computing, and other information communication technologies to track and transmit crop status as well as important environmental aspects, which is a crucial component of agriculture. (Geng L. and Dong T., 2017).
A farmer can detect plant conditions, track crop development, and access plant damage using crop monitoring technologies. Crop monitoring solutions are used to track changes that could affect farms, including environmental hazards like flooding or the expansion of water bodies, geohazards, biological hazards, and changes in land use in real-time and prompt an immediate response.
A farmer might observe environmental changes on his farm, do a risk assessment, do an early growth assessment, and acquire effective agricultural knowledge throughout a crop life cycle using crop monitoring technology including active and passive sensors, drones, and satellites. Farmers and other interested parties may view and monitor their crops remotely.
According to Ali M. et al, 2022 using satellite data with high resolution is the most powerful method to monitor crop parameters and other critical crop conditions. Panwar E. et al. 2022, established a methodology for monitoring sugarcane crop biochemical characteristics utilizing satellite-derived variables such as vegetation indices and images of the field captured from drones, offering the advantage of regular monitoring of the crop field every day without being in the field.

Yield prediction provides an estimate of crop output, which will help farmers, policymakers and investors plan properly, machine learning has been widely used for crop yield prediction (Saranya M. et al., 2020). using an agricultural dataset that includes various features such as temperature, soil properties, rainfall etc, crop yield can be accurately predicted and help the farmer decide on what type of crop will best achieve maximum yield with the resources available.
Jeong S. et al, 2022 Proposed a novel approach for early prediction of rice yield at pixel scale using satellite images by combining the synthetic use of a crop model and a Deep Learning model for different agricultural systems throughout south and north Korea. The study achieved good performance with a root mean square error of 0.605 Mg per ha and was able to predict the crop yield even in regions where data acquisition is difficult such as North Korea, they recommended that their study could be improved by transferring the method to various regions.

2.3 Livestock management

Livestock management involves agricultural practices performed to improve the production of livestock; Livestock farming practises have begun including the use of technological equipment in breeding and maintenance of livestock,
Modern technology such as artificial intelligence, sensors, cloud computing, cameras and microphones are used to monitor and perform a certain task in livestock management with minimal human effort (Singh W. et al., 2020),
Data gathered and analysed from livestock farms will help farmers properly manage their livestock, Task such as livestock tracking, automatic feed, automatic milking of dairy animals, etc can be performed, this will improve the quality of product, improve animal health and welfare, minimise the cost of operation, and improve efficiency (Micle D.E et al., 2021; Foulkes J., 2013)

2.4 Advantages and Limitations of AI and machine learning in agriculture

The use of machine learning and artificial intelligence in agriculture has improved over the years, revolutionising how agricultural data and information are being processed, AI and ML would soon play a significant role in the production, storage and marketing of agriculture and agricultural products.
The management and sustainability of resources are one of the key benefits of AI and machine learning. AI technology can help farmers manage their limited resources efficiently to achieve the maximum output. A farmer may use artificial intelligence (AI) to get a deeper understanding of the agricultural production process and crop management operations, analyse market demand, anticipate the pricing of his crops, choose the best times to plant and harvest and make other data-backed decisions.
One of the key benefits of ML approaches is their capacity to solve complex non-linear problems on their own while utilising datasets from several sources. (Chlingaryan A. et al 2018) With the large availability of data on farms today machine learning discovers formerly unknown knowledge and identifies relationships in datasets by analysing millions of data within a short time to improve agriculture without being programmed. ML's ability to solve logical problems is well suited to the agriculture industry.
The issue of labour shortages can be resolved with machine learning and artificial intelligence through automation; another benefit of AI and ML is risk management and reducing human error in

agricultural tasks. Artificial intelligence and machine learning can increase productivity and effectiveness on the farm with little error or risk, which will increase the farm's yield and output. (

2.4.1 Limitations of artificial intelligence and machine learning

The agricultural sector has gone through a transformation, AI-powered farm technologies have transformed the way we solve farming challenges bringing forward a new era of farming, however, this new era comes with limitations and challenges.
A significant barrier to implementing AI and Machine learning techniques in farming is the adoption rate of technology by farmers, most farmers, particularly those from rural areas, have little to no familiarity with technology or technical equipment, implementing Artificial intelligence technology challenges. Effectively using AI and ML technology involves the installation and Maintenace of technological equipment and infrastructure. farmers would have to hire experts or learn new skills to be able to productively utilize AI and ML.
Cost is a major factor in AI and machine learning projects; the cost of collecting, storing, processing, and analysing data for machine learning could be quite high for the farmer. There are also limitations of data availability (Ahmed M. U. & Hussain I. 2022), data is an essential entity in machine learning and artificial intelligence. The acquisition of data is a crucial limitation of machine learning applications in agriculture, acquiring the relevant farm data requires resources and skills that a farmer may not have, and there are no open-source platforms for data sharing and exchanging between farmers. (Van evert et al., 2017)
Machine learning has technical limitations (Paudel D. et al., 2021), Some certain data has to be extensively pre-processed, to be able to properly harness the power of AI and machine learning in agriculture a farmer would have to pre-processed data to remove irrelevant and redundant information, this would require advanced pre-processing techniques, a lot of computational power and technical know-how.

2.5 Literature of related current research on improving crop yield

A systematic literature review (SLR) was conducted to gain an overview of what has been done on the application of machine learning in agricultural yield prediction; this will help me identify potential research gaps on the subject. Throughout the research, all relevant publications were retrieved from search engines, journals, conference papers, and electronic databases such as Google Scholar, ResearchGate, and ScienceDirect, among others.

During the research, the goal was to better understand the approaches, datasets, and features that should be used when building models for crop yield prediction using various parameters. literature suggests that different methods and algorithms have been used by different researchers; it also suggests that several datasets were used with varying features to improve the production of crop yield rate and predict crop yield (Medar, R. et al., 2019), the research being presented and integrated here relates to the research question in our study

The performance of the AI model for crop yield prediction depends on the algorithm and dataset used in the experiment, (Kim N. et al., 2019) deep learning can effectively overcome the drawbacks of traditional methods for the prediction of crop yield forecasting (Khaki and Wang, 2019). Wang L. and Saeed K. 2019, used the 2018 Syngenta crop challenge dataset to predict crop yield performance, The dataset, consists of 2,267 maize hybrids planted in 2,247 locations between 2008 and 2016 across the United States and Canada. Using a deep neural network, they demonstrated superior performance with a root-mean-square error (RMSE) of 12%.

Some studies have created machine learning models that generate predictions using large datasets, Kale s., et al, 2019 implemented a model that helps farmers predict the crop yield and success rate of crops using neural network regression modelling. The dataset obtained from the Indian government had variables such as area, crop, state, district, season, year, and production which were spilt into a test and train set using an 80:20 assignment. The accuracy of the data was calculated based on predicted and actual values by comparing them, their model achieved an accuracy of 82% which can be improved by adding more layers and parameters.

Nigam A, et al, 2019 Presented various machine learning classifiers like XGBoost, logistic regression, Linear Regression, Artificial neural network and random forest for predicting crop yield to help farmers in choosing the best crop to grow using temperature, rainfall, season and area as variables with a dataset obtained from the Indian government, the result obtained shows that Random Forest achieved the highest accuracy when all parameters are combined. In another study, Kumar, Y. J.N et al., 2020 implement a system to predict crop production from past data obtained from the Kaggle website; the dataset is composed of 3,101 instances with 5 features including temperature, rainfall, humidity, PH and crop name. The dataset was split into 80: 20 ratios for testing and training. The accuracy of the model was predicted using different algorithms was concluded that the Random Forest algorithm produces the highest accuracy.

To estimate agricultural yields across multiple districts in Bangladesh, Ahamed A.M. S., et al., 2015 considered the impact of climatic variables (rainfall, humidity, temperature, and sunshine), biotic feature (ph., soil salinity), and the area of production. This research dataset was gathered from BARI (Bangladesh Agricultural Research Institute). Various clustering techniques were applied to divide the regions, and then classification techniques were used to obtain crop yield prediction. These data mining approaches produced results that ranged from 90 to 95% accuracy on linear regression, KNN and Neural Networks; they advised that geographic analysis be incorporated into their model to boost performance.

Using the long-term agromet-spectral dataset for the years 2001 to 2017, Prasad N.R. et al., 2021 built a model. The study used the Random Forest method to put their model into practice and forecast cotton yield for the Indian state of Maharashtra in the months of September, December, and February. The dataset was obtained from the database of economics and statistic, Ministry of Agriculture, India. For the months of September, December, and February, the observed and predicted yield comparison yielded an R2 of 69%, 60%, and 39% respectively.

Mishra S. et al., 2018 Focused on applying a crop yield prediction system by using data mining techniques on agricultural datasets obtained from Kaggle, with features which include Sr. No, district name, year, crop, area and production. J48, LWL, LAD tree, and IBK were among the classifiers employed, and their accuracy scores were 78.145%, 66.225%, 62.251%, and 80.794%, respectively.

Crop production prediction might be accomplished using data mining approaches, and some researchers have been found to do so (Beulah R., 2019). A user-friendly online website called "crop Advisor" was created by Veenadhari S. et al. in 2014 to forecast crop production based on the effect of climate conditions. The study's crop choices are based on the major crops grown in the district of Madhya Pradesh, which include soybean, paddy, maize, and wheat. The agricultural production of the chosen crop in the chosen district was predicted using secondary source data that included factors like rainfall, temperature, cloud cover, etc. Decision rules and a decision tree based on the C4.5 algorithm were used to attain accuracy levels exceeding 75%. Any user may use this approach to forecast crop production by entering the location's climatic information on the website.

Shah V. and Shah P. 2018 used two types of datasets in their study. The dataset was obtained from the department of meteorological centre at the Anand Agriculture University (AAU), and the yield data was taken from four different yearbooks of the Gujarat State Directorate of Agriculture. The dataset covered the years 2006 to 2013 and consisted of 20 attributes after pre-processing, including environmental attributes (rainfall, temperature, wind speed, pressure, and humidity), soil attrition, and other variables. In their research, four distinct algorithms were employed. The model was assessed using RMSE and contrasted with other models of four distinct alternatives. The model was examined using multiple linear regression, regression tree, artificial neural network, and K-nearest neighbour. The k-nearest neighbour prediction model performs better in both training and validation for predicting ground nut crop production

To estimate agricultural output using machine learning, Joshua S.V. et al. executed a study in 2022. The dataset utilised for the study included 470 samples from 28 districts across a variety of Tamil Nadu during the Kharif. The information includes rainfall, temperature, fertiliser (nitrogen, phosphorous, and potash) in kg, and more for the season (June-Sept) over a period of 18 years from 1998 to 2015 over a field size of 1 hectare of paddy fields. The Tata-Cornell Institute for

Agricultural and Nutrition (TCI), the regional meteorological centre in Chennai, the Tamilnadu statistics department, and the Tamilnadu agricultural department provided the data that was utilised. The statistical model of multi-linear regression served as the benchmark for predicting agricultural production. Support vector machines, general regression neural networks, and backpropagation neural networks were used to develop the model, which was then assessed using a variety of parameters. RMSE, NMSE, MAE, and so on. GRNN was more accurate, 97%, with an NMSE (normalised mean squared error) of 0.03.

Sajja G.S. et al., 2021 presented a machine learning-based framework for crop yield prediction using crop details. The framework is based on a crop yield data set with 750 instances, which includes attributes like year, region name, crop (cotton, groundnut, jowar, rice, and wheat), season (Kharif, rabi, summer), area (in hectares), production (in tonnes), average temperature, rainfall, soil, ph. value, soil type, etc The accuracy of this research was 70%, 85%, and 95%, respectively, utilising ID3, Random Forest, and Support Vector Machine.

To create a trained model to identify patterns among data and produce predictions for four important crops, including potatoes, rice, wheat, and maize, Pant J. et al., 2021 employed machine learning techniques. The input fields for the dataset provided by FAOSTAT (food and agriculture organisation of the United Nations) were as follows: item collected, country, item year beginning in 1990 through 2016, and yield value for these years. The dataset was pre-processed, and then 70% was allocated to training and 30% to testing. Support vector machines, decision trees, random forests, and gradient-boosting regressors were used to build the model. When compared to other algorithms, the decision tree regressor had the best accuracy of 96%.

Numerous researchers have looked at artificial intelligence (AI) models like random forest (RF), support vector machine (SVM), and deep neural network (DNN) as effective approaches to estimating agricultural productivity (Kim N. et al 2020). Pandith V. et al., 2020 used data obtained from the Department of Agriculture, Talab Tillo, Jammu; it contained soil samples from farms in the Jammu area. The goal of this study was to predict Mustard Crop production from soil analysis. This model was implemented using five supervised machine learning techniques: K-Nearest Neighbour (KNN), Naive Bayes, Multinomial Logistic Regression, Artificial Neural Network (ANN), and Random Forest. Performance was assessed using several metrics, including accuracy and f-score. The best performance was attained by random forest with an accuracy of 94.13%.

s/n	Researcher(s)	Dataset used	Algorithm Employed(s)	Result
1	Wang L. and Saeed K. 2019	the 2018 Syngenta crop challenge dataset	Deep Neural network	Root-mean-square-error (RMSE) of 12%
2	Kale s., et al, 2019	Indian government agriculture dataset	Neural network	82% accuracy

3	Nigam A, et al, 2019	Indian government agriculture dataset	XGBoost, Logistic regression, Linear Regression, Neural network and Random Forest	Random forest classifier had the Highest accuracy of 67.80%
4	Kumar, Y. J.N et al., 2020	Obtained from Kaggle website	Different algorithm	Random forest had the highest accuracy
5	Ahamed A.M. S., et al., 2015	Obtained from BARI (Bangladesh Agricultural Research Institute)	linear regression, KNN and Neural Networks;	90 to 95% accuracy
6	Prasad N.R. et al., 2021	Obtained from the database of economics and statistic, Ministry of Agriculture, India.	Random forest	R2 of 69%, 60%, and 39%
7	Mishra S. et al., 2018	Obtained from Kaggle	J48, LWL, LAD tree, and IBK	78.145%, 66.225%, 62.251%, and 80.794%, respectively
8	Veenadhari S. et al., 2014	secondary source data	C4.5 algorithm	accuracy levels exceeding 75%
9	Shah V. and Shah P. 2018	Department of meteorological centre at the Anand Agriculture University (AAU), and Gujarat State Directorate of Agriculture	multiple linear regression, regression tree, artificial neural network, and K-nearest neighbour.	The k-nearest neighbour prediction model performs better in both training and validation for predicting ground nut crop production.
10	Joshua S.V. et al., 2022	470 samples from 28 districts across a variety of Tamil Nadu during the Kharif	Support vector machines, general regression neural networks, and back	GRNN was more accurate, 97%, with an NMSE (normalised mean

			propagation neural networks	squared error) of 0.03.
11	Sajja G.S. et al., 2021	750 instances of their own data	ID3, Random Forest, and Support Vector Machine	he accuracy of this research was 70%, 85%, and 95% respectively.
12	Pant J. et al., 2021	FAOSTAT (food and agriculture organisation of the United Nations)	Support vector machines, decision trees, random forests, and gradient boosting regressors	the decision tree regressor had the best accuracy of 96%.
13	Pandith V. et al., 2020	Department of Agriculture, Talab Tillo, Jammu;	K-Nearest Neighbour (KNN), Naive Bayes, Multinomial Logistic Regression, Artificial Neural Network (ANN), and Random Forest.	The best performance was attained by random forest with an accuracy of 94.13%.

Table 1:Summary of Related Literature

2.6 climate change and agriculture.

Since the 1970’s an increase in the average daily temperature has been observed [], this change in temperature and cycle of weather over a long period of time is what is known as climate change. (United Nations, 2022). Climate change is a constant throughout the history of our planet and is caused by the combination of two main factors: the greenhouse effect and a rise in greenhouse gas emissions. (Ivanynya, M. et al., 2021).

The term "greenhouse effect" refers to the phenomenon whereby greenhouse gases, which include both naturally occurring gases like carbon dioxide and methane as well as synthetic gases like fluorinated gases, are trapped, heated, and reflected to the Earth's surface to warm the planet.

The second factor that contributes to climate change is the rise in greenhouse gas emissions, which was brought on by the industrial, economic, and technological revolutions as well as changes in population, burning coal, pollution from oil and gas refineries, deforestation, farming of livestock and other forms of agriculture, transportation, and other economic activities. (US EPA, 2022) The amount of greenhouse gases generated has dramatically grown as a result of these human activities. According to a 2013 assessment by the Intergovernmental Panel on Climate Change, methane emissions climbed by 150 Percent between 1750 and 2011, carbon dioxide emissions increased by 40 Percent, and nitrous oxide emissions increased by 20 Percent. (Stocker, T.F et al., 2013)

As a result of the greenhouse effect and rising greenhouse gas concentrations, an increase in temperature has been seen in the global average temperature of the near-air surface and sea surface temperatures (Morice et al. 2012). Although, while a 1-degree rise in temperature might appear insignificant, it has a significant impact on the weather and climate (Ritchie H., at al. 2020)

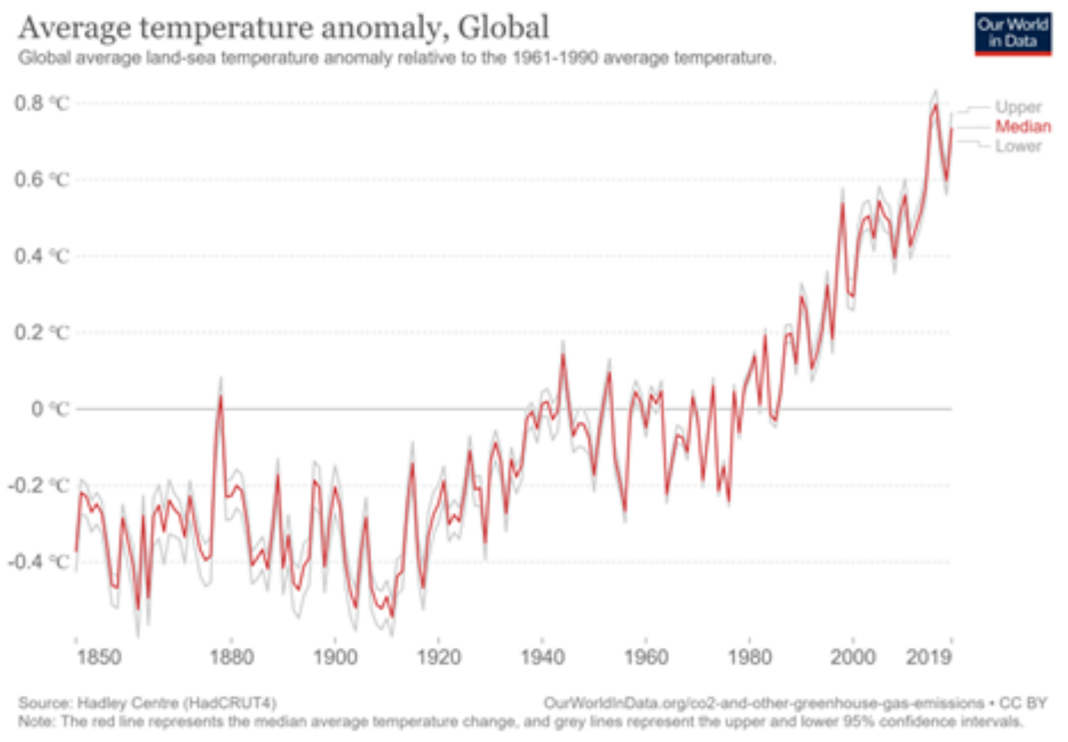

Figure 2: Average temperature anomaly, global (Our world in data, 2022)

Climate change is a looming problem that needs to be addressed, the rise in temperature causes an increase in evaporation which leads to increased frequency and intensity of extreme weather conditions, effects of climate change occurring gradually has become increasingly visible examples include rising sea levels, changing rainfall patterns, melting of the ice caps, storms, drought, wildlife fire, flooding, global temperature rise, floods, landslides, hurricanes, tornadoes, wildfires, desertification and loss of biodiversity (Rolnick D., 2022),

To combat climate change, they must be a rapid reduction in global greenhouse gas emissions, and adaptation measures are taken to prepare for inevitable consequences (Rolnick D. et al., 2022) effect of climate change in agriculture, the economy and general health of human population is raising global concern.

2.6.1 Climate change impact on agriculture.

Numerous factors shape and drive the agricultural sector, climate change has been recognised to have a significant implication on Agriculture, agriculture is highly vulnerable to climatic conditions and is recognised as a primary factor for agricultural production (FAO, 2022).
The effects of climate change vary by region of the world and will continue to affect the output of crops, animals, and ecosystems. Concerns about the future of global food production are growing due to the projected and observed effects of climate change on agriculture and food security (Nyasimi M., 2017), understanding the impact of climate change on agriculture will help direct where, when and how adaptation should proceed. (Mendelsohn R., 2009)
The combined effect of climate change has a direct impact on food production across the globe; Increase in mean temperature, changes in rain patterns, desertification, drying of water bodies, flooding, worsening soil conditions, affecting the cultivation of crops, reduce crop yield and viability of the agricultural product, climate change effect also encourages pest and disease to thrive

2.6.2 Agriculture's contribution to climate change.

Agriculture affects and is affected by climate change, livestock, Food production, processing, and consumption release greenhouse gases into the atmosphere. Agriculture and food production is a major contributor to climate change, they are a wide range of sources for Greenhouse gas (GHS) emissions in agriculture including, crop cultivation, livestock farming, forestry, food processing synthetic fertilizers, etc ().
Farming releases a significant amount of methane and nitrous oxide and other greenhouse gas. According to projections from the FAO, GHS emissions from forestry, agriculture, and fisheries have quadrupled over the previous 50 years and may rise by another 30% by 2050. (FAO, 2014), Livestock production remains a major source of GHGs, especially CH_4, contributing about 19-29% of total greenhouse gas (GHG) emissions, (). Crop and livestock production saw an increase in emissions from 4.7 billion tonnes of carbon dioxide in 2001 to over 5.3 billion tonnes in 2011. (Lal R., 2021; Tubiello F.N. et al., 2014)
A reduction of greenhouse gas emissions from agriculture would have numerous benefits to the ecosystem, however, it remains quite challenging, this can be achieved by reducing the usage of synthetic fertilizer, integrating innovative technology into production to reduce GHGs emissions and encouraging the usage of renewable energy in agriculture. (EEA, 2021)

2.7 climate-smart agriculture practises

Affected by the effects of climate change, agricultural viability and productivity are at risk from high temperatures, droughts, floods, changes in rainfall patterns, extreme weather events, the spread of various pests and diseases, and a rise in the frequency of forest fires ().

Agriculture is one of the most important sectors of the economy, plays a vital role in ensuring food security, and contributes significantly to the GDP of many nations.

The United Nations Framework Convention on Climate Change first used the phrase "climate-smart" in 2010, and organisations all around the world have since embraced the idea as a response to climate change's effects (Brohm K.A. and Klein S., 2020).

Climate-smart Agriculture (CSA) is a strategy for reshaping agricultural production in light of the effects of climate change while also addressing the problems of fulfilling the rising demand for food, fibre, and fuel (Lipper et. Al.,2014; Steenwerth, K.L., et al., 2014)

climate-smart agriculture helps farmers adapt to and mitigate the effects of climate change, increased resilience to climate variability and withstand extreme weather conditions; it is an integrated approach to managing landscapes, crops, and livestock to address interconnected concerns of food security and climate change, three objectives are sought after by CSA practice. Enhancing agricultural yields, improving resistance to climate change, and reducing greenhouse gas emissions are all goals. This is accomplished by combining several environmentally friendly techniques to address the unique climate problems faced by a particular farming community (FAO, 2021; world bank, 2022)).

CSA Instead of being a collection of universally applicable practises, it is a strategy that incorporates various elements into particular local situations. Creating a CSA plan includes recognising the primary issue (situation analysis), creating and prioritising solutions and design plans, implementing the solution, and evaluating it. This answer may take many different forms, such as Brazil’s low-carbon agriculture plan, drought-tolerant maize, and agricultural management techniques, Consider planting shade trees and conserving water. (Nerger M., 2022; Palombi, L., and Sessa, R., 2013)

A variety of techniques and technologies are used in the CSA intervention, which has been effectively applied all over the world to address the agricultural issue brought on by climate change (FAO, 2021) by carefully managing of resources like soil, water and biodiversity.

3 METHODOLOGY

3.1 Proposed system

The term "methodology" refers to the examination of the strategies, ideas, and methods employed in research. Improving crop yield prediction using machine learning involves using data and analytical techniques to develop more accurate models for predicting crop yield. To improve crop yield predictions, several approaches can be used.

This study implements an approach that improves crop yield prediction accuracy by improving the accuracy of the key attributes that contribute to crop yield, by improving the accuracy of the individual attributes that contribute to crop yield, researchers can build more accurate and reliable crop yield prediction model, this can help farmers make more informed decisions about the individual farming practices and will ultimately lead to higher crop yields

This study implemented a web application framework developed in python using the Streamlit framework, the web interface named “Agro-predict “, which would serve as a portal that farmers could use to generate predictions using their data as input.

The proposed system contains five modules, namely crop recommendation modules, this module predicts the most suited crop to grow based on soil nutrient and environmental factors. The fertilizer recommendation, this module gives farmers information on the kind and quantity of fertilizer to apply to their crops based on the quality of nitrogen, potassium, and phosphorus in the soil, and soil type. The plant disease module detects and diagnoses plant disease using plant images. The rainfall prediction module will help predict rainfall using historic rainfall data. The weed detection module helps farmers detect weeds properly.

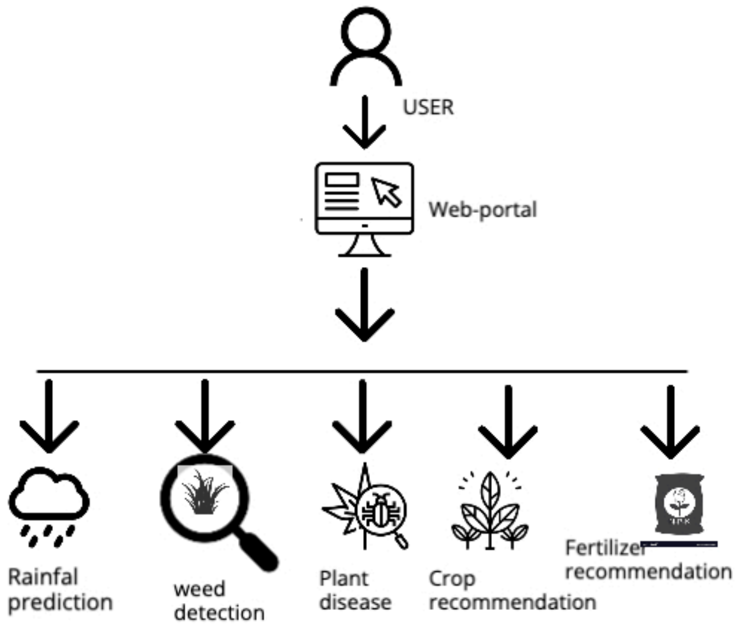

Figure 3: Proposed system

3.2 Crop recommendation

A farmer's decision about which type of crop to grow is usually influenced by his knowledge of a crop, profit, and other irrelevant factors (Lynch, L., & Brown, C. 2000), without considering the best crop suited for his land based on nutrients available in the soil.

To fulfil growing demand and better the quality and quantity of food while mitigating the consequences of climate change, traditional agriculture has been turned into modern agriculture which uses contemporary technology in farming which includes IoT sensors used to gather data like nutrients levels of the Soil, and temperature, it also includes artificial intelligence, blockchain, cloud computing and robotics etc. These technologies improve overall efficiency in farming and maximize food productivity (Thilakarathne N. et al., 2022).

Soils play a critical role in the production of food; it provides the roots with essential nutrient Plant requires to grow, Soil is a major source of nutrient, Farmers usually employ farming strategies such as crop rotation to take adequate advantage of the nutrient present in the soil without relying on fertilizers (Kumar A. et al., 2020).

Any crop needs the proper amount of certain nutrients to develop and thrive, as well as the ideal climate, Crop recommendations using machine learning involve the usage of soil nutrient data and climatic parameters to help farmers maximize agricultural yield by recommending appropriate crops to be planted.

3.2.1 Dataset

For the crop recommendations module, the dataset contains soil-specific attributes, and environmental factors, the dataset contains 8 features Nitrogen(K), Phosphorus (P), Potassium(K), Temperature (°C), Rainfall, Ph value, humidity, and the target attribute "label" which contains crops like rice, wheat etc.

s/n	features	Importance
1	Nitrogen(K)	Responsible for photosynthesis
2	Phosphorus (P)	Responsible for storing and transfer of energy
3	Potassium(K)	Associated with the movement of water, nutrients, and carbohydrates in plant tissue
4	Temperature (°C)	Influence photosynthesis, germination, growth, and flowering
5	humidity	Influences growth levels and overall health of a crop
6	Ph value	Affects the amount of nutrients available to a crop
7	Rainfall	The main source of water

Table 2:Dataset features for crop recommendation

The dataset used for training and evaluation was acquired from Kaggle (an online data repository) (Sharma S., 2021). The dataset is licensed under Attribution 3.0 IGO (CC BY 3.0 IGO) which allows the data to be shared, adapted, and used for research purposes (Sharma S., 2021)

3.2.2 Data Pre-processing.

Data preparation is the procedure used to get a dataset ready for analysis. Data pre-processing is a crucial stage in the analysis of the data; it aims to improve the accuracy and dependability of the findings as well as make the data better suitable for analysis.

In this study, data pre-processing was done in python Jupiter notebook, the required libraries were installed and then Imported before the pre-processing can be done.

The dataset downloaded was named crop recommendation, the dataset consisted of 2,200 rows and 8 columns. and was imported using pandas, I also looked at the first 10 rows.

```
#step 2: Importing the dataset
data =pd.read_csv("Crop_recommendation.csv")
```

```
#A look at the first ten.
data.head(10).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
```

N	P	K	temperature	humidity	ph	rainfall	label
90	42	43	20.879744	82.002744	6.502985	202.935536	rice
85	58	41	21.770462	80.319644	7.038096	226.655537	rice
60	55	44	23.004459	82.320763	7.840207	263.964248	rice
74	35	40	26.491096	80.158363	6.980401	242.864034	rice
78	42	42	20.130175	81.604873	7.628473	262.717340	rice
69	37	42	23.058049	83.370118	7.073454	251.055000	rice
69	55	38	22.708838	82.639414	5.700806	271.324860	rice
94	53	40	20.277744	82.894086	5.718627	241.974195	rice
89	54	38	24.515881	83.535216	6.685346	230.446236	rice
68	58	38	23.223974	83.033227	6.336254	221.209196	rice

Figure 4:data importation for crop recommendations

Pre-processing of the data was done firstly by handling the missing data, handling the missing data refers to the process of dealing with missing or incomplete values in a dataset. Missing values were checked using the ISNA() function of pandas, and I found that the dataset did not contain any missing or incomplete values. Label encoding was then carried out on the dataset

Label encoding is a way to convert categorical variables (variables that take on a limited, fixed number of values) into numerical values. This can be useful for certain types of models that only accept numerical input and can also improve the performance of the model in some cases. In this dataset, I performed label encoding on the categorical data to convert it to numerical values, which can improve the performance of the algorithm.

```
#step 3: Handling the missing Data
#checking the total missing value in each features
print('Total missing value :')
data.isna().sum()
```

```
Total missing value :

N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

Figure 5:finding any missing value for crop recommendations

The dataset was subsequently splitted into a training set and a testing set, an essential step in assessing the effectiveness of a machine learning model is dividing the data into training and testing sets. The training set is used to develop the model, while the testing set is used to assess how well it performs, in this module, I divided the dataset using the train_test_split function of Sklearn, I allocated 75% of train the data and 25% to testing.

```python
#Step 5: splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(data.drop('label', axis=1),
data['label'], test_size=0.25, random_state=42)
```

```python
#printing the shape of the training set and test dataset
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1650, 7)
(550, 7)
(1650,)
(550,)
```

```python
#printing the training set
print(X_train)
```

```
        N    P    K  temperature   humidity        ph    rainfall
564    22   36   16    30.581395  50.771481  8.184229   64.585596
916    18   27   41    22.365094  92.308824  7.175344  104.821633
1700   61   68   50    35.214628  91.497251  6.793245  243.074507
436    26   72   22    28.767949  37.577921  4.674942   91.720849
1555    2  140  197    22.697801  92.822234  5.534567  105.050823
...   ...  ...  ...          ...        ...       ...         ...
1638   10    5    5    21.213070  91.353492  7.817846  112.983436
1095  108   94   47    27.359116  84.546250  6.387431   90.812505
1130   11   36   31    27.920633  51.779659  6.475449  100.258567
1294   11  124  204    13.429886  80.066340  6.361141   71.400430
860    32   78   22    23.970814  62.355576  7.007038   53.409060

[1650 rows x 7 columns]
```

Figure 6 : test-train split for crop recommendations module

3.3 Fertilizer recommendation module

A fertilizer is a material that is applied to soil to provide plants with nutrients like nitrogen, phosphorus, and potassium. Fertilizers can be applied in a range of forms, such as liquid, granular, or pelletized, and are often manufactured from inorganic or organic components. The usage of fertilizers can enhance plant health and growth, as well as crop quality and output.

A fertilizer recommendation system gives farmers advice on the kind and quantity of fertilizer to apply to their crops. The advice is based on several variables, including the nutrient available in the soil, the kind of crop, the soil, and the weather. A fertilizer recommendation system's objective is to assist farmers in utilizing the proper quantity of fertilizer at the right time to increase crop yields while minimizing environmental damage (Snyder, C. S. et al., 2009). In addition to enhancing the quality of their crops and the health of their land, this will help farmers save money and resources.

Machine learning-based fertilizer recommendation systems would use data analysis and prediction algorithms to offer farmers individual personalized fertilizer recommendations.

3.3.1 Dataset.

For the fertilizer recommendation module, the dataset used was acquired from Kaggle, (an online data repository) Abhishek G.D., 2019. It consists of 8 features and a target class called fertilizer name, the features include potassium, and phosphorus temperature, humidity, soil type, moisture, crop type, and nitrogen. Abhishek G.D., 2019.

3.3.2 Data Pre-processing.

Pre-processing of the fertilizer recommendation data was done by importing the required libraries and then first previewing the first 10 rows using pandas.

```
#step 2: Importing the dataset
data =pd.read_csv("Fertilizer Prediction.csv")
```

```
##A look at the first ten row.
data.head(10).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
```

Temparature	Humidity	Moisture	Soil_Type	Crop_Type	Nitrogen	Potassium	Phosphorous	Fertilizer_Name
26	52	38	Sandy	Maize	37	0	0	Urea
29	52	45	Loamy	Sugarcane	12	0	36	DAP
34	65	62	Black	Cotton	7	9	30	14-35-14
32	62	34	Red	Tobacco	22	0	20	28-28
28	54	46	Clayey	Paddy	35	0	0	Urea
26	52	35	Sandy	Barley	12	10	13	17-17-17
25	50	64	Red	Cotton	9	0	10	20-20
33	64	50	Loamy	Wheat	41	0	0	Urea
30	60	42	Sandy	Millets	21	0	18	28-28
29	58	33	Black	Oil seeds	9	7	30	14-35-14

Figure 7: previewing dataset for fertilizer recommendations module

The dataset used contained no missing or incomplete values and had a total of 99 rows and 9 columns. To improve the performance of the algorithm, I performed label encoding on the categorical data to convert it to numerical values. The dataset was split into testing and training sets using 25% for testing and 75% for training, this was archived by using the test_train_split function of Sklearn.

```
#label encoding of the categorical data
encoder = OrdinalEncoder()
encode_cols = ['Soil_Type', 'Crop_Type', 'Fertilizer_Name']
encoder.fit(data[encode_cols])
data[encode_cols] = encoder.transform(data[encode_cols])
```

```
print(data)
```

```
    Temparature  Humidity  Moisture  Soil_Type  Crop_Type  Nitrogen  \
0            26        52        38        4.0        3.0        37
1            29        52        45        2.0        8.0        12
2            34        65        62        0.0        1.0         7
3            32        62        34        3.0        9.0        22
4            28        54        46        1.0        6.0        35
..          ...       ...       ...        ...        ...       ...
94           25        50        32        1.0        7.0        24
95           30        60        27        3.0        9.0         4
96           38        72        51        2.0       10.0        39
97           36        60        43        4.0        4.0        15
98           29        58        57        0.0        8.0        12

    Potassium  Phosphorous  Fertilizer_Name
0           0            0              6.0
1           0           36              5.0
2           9           30              1.0
3           0           20              4.0
4           0            0              6.0
..        ...          ...              ...
94          0           19              4.0
95         17           17              0.0
96          0            0              6.0
97          0           41              5.0
98          0           10              3.0

[99 rows x 9 columns]
```

Figure 8: Label encoding for fertilizer recommendations module

```
#Step 5: splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(data.drop('Fertilizer_Name', axis=1),
data['Fertilizer_Name'], test_size=0.25, random_state=42)
```

```
#printing the shape of the training set and test dataset
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(74, 8)
(25, 8)
(74,)
(25,)
```

```
#printing the training set
print(X_train)
```

```
    Temparature  Humidity  Moisture  Soil_Type  Crop_Type  Nitrogen  \
33           36        68        38        4.0        0.0         7
9            29        58        33        0.0        5.0         9
81           30        60        40        4.0        4.0        41
11           31        62        48        4.0        3.0        14
65           36        68        62        3.0        1.0        15
..          ...       ...       ...        ...        ...       ...
60           28        54        41        1.0        6.0        36
71           31        62        32        3.0        9.0        39
14           26        52        31        3.0        2.0        14
92           36        68        41        3.0        2.0        41
51           36        68        33        0.0        5.0        13

    Potassium  Phosphorous
33          9           30
9           7           30
81          0            0
11         15           12
65          0           40
..        ...          ...
60          0            0
71          0            0
14          0           41
```

Figure 9: Dataset split for fertilizer recommendations module

3.4 Plant disease

Plant disease is a condition that affects the health and growth of plants and is caused by a pathogen such as a virus, bacteria, fungus, or parasite. Plant diseases can reduce a plant's ability to photosynthesize, reproduce, or resist environmental stresses, which can affect its yield and overall health. To improve crop yield, it is important to prevent and control plant disease.
Farmers can use a range of strategies, such as planting disease-resistant crops and applying pesticides, to prevent and control plant disease. However, traditional methods of recognizing plant disease, such as observing the affected plants and making an educated guess about the disease, are unreliable and can result in the indiscriminate use of chemicals.
Machine learning can be used to detect and diagnose plant disease by analysing images of healthy and unhealthy plants. This method involves training algorithms using a large dataset of labelled plant images to identify patterns and features associated with specific diseases.

3.4.1 Dataset

The dataset used for the plant disease module came from Kaggle and was made up of two distinct datasets that were combined into one (Khan, 2022). The majority of the photos were initially taken from the plant Village dataset, which was first made available on Mendeley. The photographs were gathered from lab scenes and in the wild settings, and the dataset is published under the CC0: public domain. It comprises 20k images of tomato leaves with 10 disease classes and 1 healthy class, the disease includes Late_blight, Early_blight, Septoria_leafspot, Tomato_Yellow_LeafCurlVirus, Bacterial_spot, Target_Spot, Tomato_mosaic_virus, Leaf_Mold, Spider_mites Two-spottedspider_mite, Powdery Mildew. Khan, 2022.

3.4.2 Data pre-processing

The data for this machine learning model was already organized into a structured directory and had been properly formatted. Pre-processing was performed by loading the images, extracting relevant features, and preparing the data for training and testing.
To extract the features, the images were looped through and resized to a fixed size of 32x32 pixels. They were then converted to grayscale and flattened into a 1D array. The resulting data and labels were converted to NumPy arrays.
The data was then split into training and testing sets using the train_test_split function from sklearn.model_selection. This is a common practice in machine learning, as it allows the model to

be evaluated on unseen data. The training set is used to train the model, while the testing set is used to evaluate its performance.

```
#previwing random images from the folder path
show_images(11, folder_paths)
```

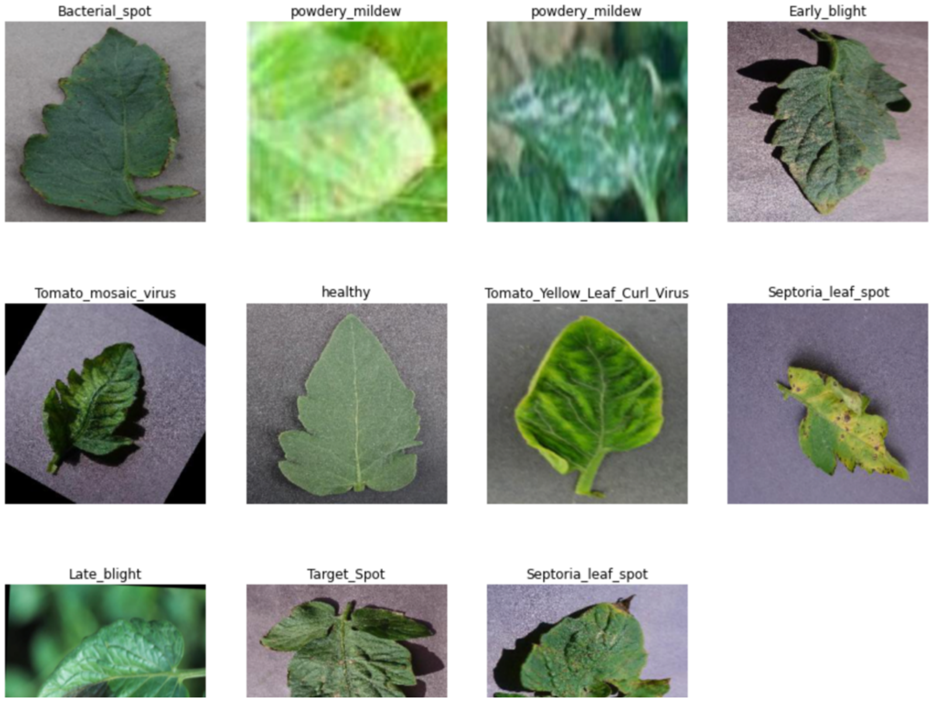

Figure 10: A preview of plant disease images

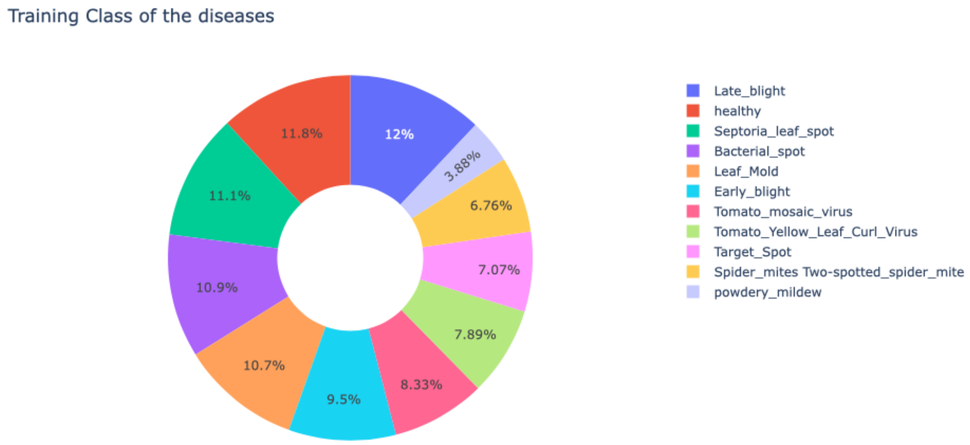

Figure 11: visualization of the plant disease images training class

3.5 Rainfall prediction

Rainfall is one of the key factors that can affect crop yield, as plants need a certain amount of water to grow and produce a good harvest. In general, crops that require a lot of water, such as rice and corn, are more sensitive to changes in rainfall than drought-tolerant crops, such as wheat and barley.

Machine learning algorithms can be used to help predict rainfall and its impact on crop yield. By analyzing historical data on rainfall and crop yields, as well as other factors that can affect crop growth, such as temperature and soil moisture, machine learning algorithms can learn to make accurate predictions about future rainfall and how it will impact crop yield.

These predictions can be used by farmers to help them make informed decisions about when to plant and harvest their crops, as well as how much water and other resources they should allocate to their crops and manage irrigation in order to maximize their yield. In this way, machine learning can help farmers optimize their operations and improve their crop yield.

Additionally, rainfall prediction can also help farmers prepare for potential risks of flooding or drought, which can also impact crop yields. By staying informed about rainfall patterns, farmers can take steps to mitigate potential risks and maximize crop yields.

3.5.1 Dataset

The dataset used for the rainfall prediction module was obtained from Kaggle, Young J., 2020. The dataset consists of 10 years of daily weather observation from many weather stations across Australia. The dataset is made of 22 features and 1 target class, the features include temperature, rainfall, evaporation, sunshine, humidity, wind speed, etc. Young J., 2020.

3.5.2 Data pre-processing

Pre-processing was done by first importing the required libraries, which are necessary for pre-processing, the dataset downloaded was named WeatherAUS and was imported using pandas, I also looked at the first 10 rows.

```python
#step 2: Importing the dataset
data =pd.read_csv("weatherAUS.csv")
```

```python
##A look at the first ten row.
data.head(10).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
```

Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed
2008-12-01	Albury	13.400000	22.900000	0.600000	nan	nan	W	44.000000	W	WNW	20.000000	24.00
2008-12-02	Albury	7.400000	25.100000	0.000000	nan	nan	WNW	44.000000	NNW	WSW	4.000000	22.00
2008-12-03	Albury	12.900000	25.700000	0.000000	nan	nan	WSW	46.000000	W	WSW	19.000000	26.00
2008-12-04	Albury	9.200000	28.000000	0.000000	nan	nan	NE	24.000000	SE	E	11.000000	9.00
2008-12-05	Albury	17.500000	32.300000	1.000000	nan	nan	W	41.000000	ENE	NW	7.000000	20.00
2008-12-06	Albury	14.600000	29.700000	0.200000	nan	nan	WNW	56.000000	W	W	19.000000	24.00
2008-12-07	Albury	14.300000	25.000000	0.000000	nan	nan	W	50.000000	SW	W	20.000000	24.00
2008-12-08	Albury	7.700000	26.700000	0.000000	nan	nan	W	35.000000	SSE	W	6.000000	17.00
2008-12-09	Albury	9.700000	31.900000	0.000000	nan	nan	NNW	80.000000	SE	NW	7.000000	28.00
2008-12-10	Albury	13.100000	30.100000	1.400000	nan	nan	W	28.000000	S	SSE	15.000000	11.00

Figure 12:Importing dataset for rainfall prediction

The dataset contained numerous missing values from each feature, the target data is rain tomorrow, and the dataset contains both numerical and categorical variables, the null values of the features were eliminated, which reduced the dataset to 56,420 rows.

The date column is registered as an object and was pre-processed by handling it with the daytime format by separating the year, Month, and day, the original date column was then deleted. The label encoding was then used to locate the categorical data and transform it into numerical data.

After pre-processing the dataset contains 24 features and the target data. The dataset was split into test and training sets, with 75% of the data used for testing and 25% used for training. The train data now contains 42,315 rows and 24 features while the test data contains 14,105 rows.

```python
#dealing with the categorical data
categorical = data.select_dtypes(include=['object'])
categorical.head()
```

	Location	WindGustDir	WindDir9am	WindDir3pm	RainToday	RainTomorrow
6049	Cobar	SSW	ENE	SW	No	No
6050	Cobar	S	SSE	SSE	No	No
6052	Cobar	NNE	NNE	NNW	No	No
6053	Cobar	WNW	WNW	WSW	No	No
6054	Cobar	WNW	NW	WNW	No	No

```python
#label encoding of the categorical data
encoder = OrdinalEncoder()
encode_cols = ['Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm','RainToday','RainTomorrow']
encoder.fit(data[encode_cols])
data[encode_cols] = encoder.transform(data[encode_cols])
```

Figure 13: label encoding for rainfall prediction module

```
#Step 5: splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(data.drop('RainTomorrow', axis=1),
data['RainTomorrow'], test_size=0.25, random_state=42)
```

```
#printing the shape of the training set and test dataset
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(42315, 24)
(14105, 24)
(42315,)
(14105,)
```

```
#printing the training set
print(X_train)
```

```
        Location  MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  \
84432        1.0     18.0     29.2       0.0          5.8       8.4
120230      16.0     13.8     22.6       3.8          0.8       6.8
85524        1.0     15.9     25.6       0.6          4.6      10.6
103201      14.0     10.4     27.7       0.0          6.4      11.1
80321       23.0      3.1     14.1       0.0          1.4       6.5
...          ...      ...      ...       ...          ...       ...
119161      16.0      8.3     19.3       9.6          0.8       8.3
140143       6.0     17.6     30.7       0.0          4.8      11.2
100693      12.0     10.3     18.2       0.2          3.2       2.0
9433         5.0     18.9     28.2       0.0          7.0       7.1
61472       18.0      5.3     17.8       0.0          2.8       6.3

        WindGustDir  WindGustSpeed  WindDir9am  WindDir3pm  ...  Pressure9am  \
84432          15.0           35.0        15.0        13.0  ...       1014.2
120230         12.0           37.0         4.0        15.0  ...       1018.9
85524           0.0           31.0         2.0         2.0  ...       1018.5
103201          3.0           50.0         5.0         5.0  ...       1022.7
80321           4.0           19.0         4.0         3.0  ...       1027.4
...             ...            ...         ...         ...  ...          ...
119161          9.0           28.0         0.0         2.0  ...       1022.1
140143          0.0           37.0         0.0         0.0  ...       1015.6
```

Figure 14:dataset split for rainfall prediction

3.6 Weed detection

To increase agricultural productivity, weed identification is important. Weed detection is the process of identifying the presence of weeds within crops. Weeds are unwanted plants that can hurt crop yield by competing with the crop for resources such as sunlight, water, and nutrients. Weed detection and control are crucial components of crop management. Farmers take measures to eliminate the weeds and safeguard their crops by spotting the presence of weeds early on, traditionally, farmers manually remove weeds and use herbicides indiscriminately, this is expensive and causes harm to humans, animals, and the environment. Weed detection techniques include physical examination, aerial imagery, and the use of specialized sensors and technologies like machine learning.

Weed detection using machine learning has been successful in the past. A dataset of quality images is uploaded and then trained to identify weed using machine learning algorithms, the model will be able to detect weed in fresh photos without manual inspection.

3.6.1 Dataset

The dataset used in this module was acquired from Mendeley, (an open data repository) the dataset consists of 15336 segments, being 3249 soils, 7376 soybeans, 3250 grass and 1191 broadleaf weeds, the dataset is licensed under the CC BY NC 3.0 dos Santos Ferreira et al., 2017.

3.6.2 Data pre-processing

The dataset used in this implementation was organized into a directory with four labels: grass, broadleaf, soil, and soybean. The load_images function was called to load the data from this directory, and the resulting data was then converted to a 2-dimensional matrix using the reshape function from the numpy library. The data was then split into training and test sets using the train_test_split function from the sklearn library, with 30% of the data being assigned to the test set.

```
show_images(11, folder_paths)
```

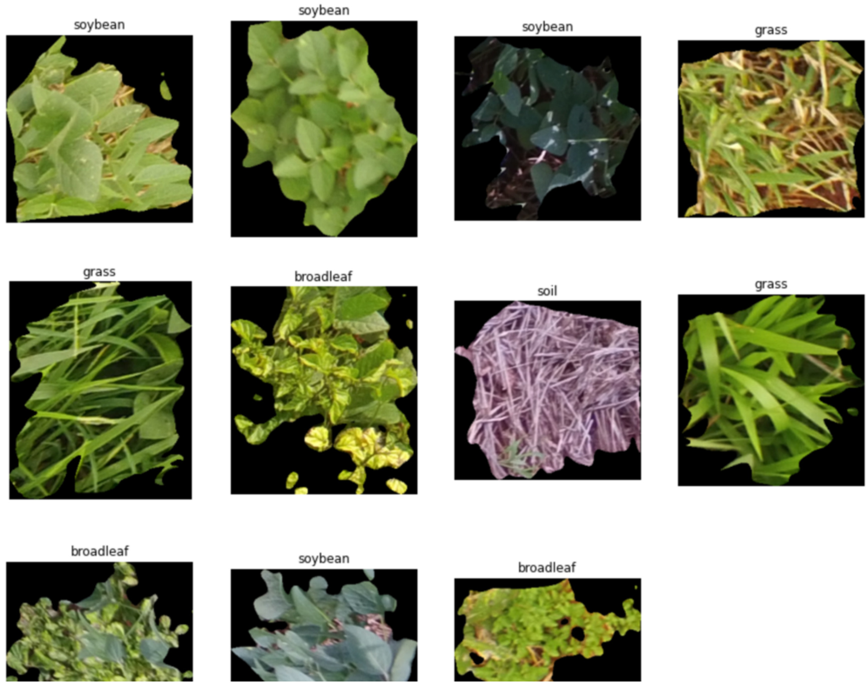

Figure 15: Displaying random images from the file path

4 IMPLEMENTATION

In machine learning, implementation refers to the process of implementing a machine learning model, which involves training the model on a dataset and using it to make predictions or perform other tasks. in this study after data preparation and preprocessing, implementation of machine learning algorithm was performed on each module to predict an outcome by learning patterns in the data, the results were then analyzed, and the performance of the algorithms was compared. The model was then deployed to a web interface that collects input from users to produce a prediction.

4.1 Algorithms Employed

In this study, I employed four machine learning algorithms for the classification tasks: XGBoost, K-neighbours classifier, random forest classifier, and support vector machines (SVMs). These algorithms were chosen based on their unique characteristics and strengths, including XGBoost's strong performance and efficiency, the simplicity and ease of implementation of the K-neighbors classifier, the random forest classifier's ability to resist overfitting, and the effectiveness of SVMs when working with high-dimensional data. I believe that these algorithms were well-suited to the classification problems at hand and played a key role in the success of this study.

4.1.1 XGBoost Classification Algorithm

Extreme Gradient Boosting is known as XGBoost. Extreme Gradient Boosting is a sort of ensemble machine learning, which combines the predictions of many models to produce more accurate predictions. Decision trees, which employ a tree-like structure to make decisions depending on the value of the input, are frequently used in predictions by these models.
The residuals (errors) between the predicted values and the actual values are calculated using gradient boosting, and a second decision tree is trained to predict these residuals. Each additional decision tree in this method is trained to anticipate the residuals of the prior tree. All of the ensemble's trees' predictions are added together to get the final result.
XGBoost Algorithm is known for its efficiency, performance, and flexibility, and has been widely used in a variety of applications such as recommendation systems, natural language processing, and computer vision. (XGBoost, 2022)

4.1.2 K-neighbours Classifier

K-Neighbors Classifier is an example of an instance-based Learning algorithm. Instance-based learning algorithms are types of machine learning algorithms that make predictions based on the similarity between new data points and the training data. In a KNN model, K, is the number of nearest Neighbors to consider when making a prediction. prediction for a new data point is made based on the "K" nearest neighbors of the point in the training data set.

One of the key benefits of the KNN algorithm is that it is simple and easy to implement. It also does not require any assumptions about the distribution of the data. KNN algorithm Is a simple and effective supervised machine learning algorithm for classification and regression tasks.

4.1.3 Random Forest Classifier

Random forest is a machine learning algorithm that is used for classification and regression tasks, it is a type of ensemble machine learning algorithm composed of decision trees.
Random forest classification combines the prediction of multiple decision trees to make more accurate predictions than any individual tree would be able to, each tree in the random forest is trained using a different subset of the training data, and the predictions of all the trees are combined.
To make predictions using a random forest classifier, the model takes an input and passes it through all the decision trees in the forest. The Random Forest algorithm's ability to handle high-dimensional data and missing values in the data is one of its main advantages. Additionally, it is largely resistant to overfitting. Ho T.K., 1995.

4.1.4 Support Vector Machines

The supervised learning algorithm known as Support Vector Machines (SVMs) can be applied to classification or regression problems. Finding the hyperplane in a high-dimensional space that maximum separates the positive and negative examples is the objective of an SVM. The SVM optimizes an objective function to locate this hyperplane, which measures how well the hyperplane can separate the classes as well as how far away it is from the nearby training instances (called the margin). The margin is the separation between the hyperplane and the closest examples. Maximizing the margin is the SVM's optimization goal.

SVMs have been widely used in many problems, including text classification and, image classification. One of the advantages of SVMs is that they can perform well even with high-dimensional data, such as text data with many features. They also have good generalization performance, meaning that they can perform well on new, unseen data. However, SVMs can be computationally expensive to train, especially for large datasets, and they do not scale well to very large datasets.

4.2 Crop recommendation Implementation

In this study, crop recommendation was treated as a classification problem, with the goal of predicting the most suitable crops to plant based on various factors. In this study, I trained three machine learning algorithms on the dataset.

The algorithm used includes XGBoost, Random Forest and KNN, the code for the machine learning algorithms used are included in Appendix A. The results of using these three machine learning algorithms are summarized in the table below. The training accuracy represents the accuracy of the predictions made by the algorithm on the training dataset, while the testing accuracy represents the accuracy of the predictions made on a separate, unseen dataset (the "test" dataset).

Overall, the results indicate that all three algorithms performed well on the training dataset, with XGBoost achieving the highest training accuracy at 100%. On the test dataset, all three algorithms also had relatively high testing accuracies, with Random Forest achieving the highest testing accuracy at 98.9091%. The module was then saved and ready to be implemented on the web interface.

Algorithm Name	Training Accuracy	Testing Accuracy
XGBoost	100%	97.8182%
Random Forest	97.8182%	98.9091%
KNN	99.3939%	96.7273%

Table 3:Testing and training Accuracy for crop recommendations

4.3 Rainfall prediction Implementation

The rainfall prediction module was implemented using three machine learning algorithms and treated as a classification problem with the goal of predicting whether it will rain tomorrow or not. The dataset was pre-processed before the prediction was carried out. The following algorithms were used to train and test the model: XGBoost, Random Forest, and KNN. The code for the machine learning algorithms used are included in Appendix A. XGBoost had a training accuracy of 87.7230% and a testing accuracy of 86.0688%. Random Forest had a training accuracy of 86.0688% and a testing accuracy of 86.3736%. KNN had a training accuracy of 90.5636% and a testing accuracy of 83.4243%. Overall, XGBoost and KNN performed well in terms of training accuracy, but the testing accuracy for these algorithms was slightly lower compared to that of the Random Forest algorithm. This indicates that the Random Forest model is more robust and performs better on unseen data. The module was then saved and ready to be implemented on the web interface.

Algorithm name	Training Accuracy	Testing Accuracy
XGBoost	87.7230%	86.0688

Random Forest	86.0688%	86.3736%
KNN	90.5636%	83.4243%

Table 4:Testing and Training Accuracy for Rainfall implementation

4.4 Fertilizer recommendation Implementation

The Fertilizer recommendation Implementation is a system that uses machine learning algorithms to provide recommendations on the type of fertilizer to be used for different crops in different regions. This module was implemented with three algorithms and treated as a classification problem. The algorithms used include XGBoost, Random Forest, and KNN, the code for the machine learning algorithms used is included in Appendix A.

In this implementation, the XGBoost algorithm achieved the highest accuracy in both the training and testing stages, with 100% accuracy in both instances. The Random Forest algorithm also performed well, achieving 100% training accuracy and 96% testing accuracy. The KNN algorithm had slightly lower accuracy rates, with a 98.64% training accuracy and a 92% testing accuracy. Based on these results, the XGBoost algorithm is recommended for use in the Fertilizer recommendation Implementation as it consistently demonstrated high accuracy rates in both the training and testing stages. The Random Forest and KNN algorithms also showed promising results but did not outperform XGBoost.

Algorithm name	Training Accuracy	Testing Accuracy
XGBoost	100%	100%
Random Forest	100%	96%
KNN	98.6486%	92%

Table 5: Testing and Training Accuracy for Fertilizer implementation

4.5 Weed detection implementation

The weed detection system uses machine learning algorithms to identify weeds in soybean crop images. It was designed as a classification problem and three algorithms were tested: XGBoost, Random Forest, and KNN, the code for the machine learning algorithms used is included in Appendix A. The training and testing accuracies for each algorithm are shown in the table below

Algorithm name	Training Accuracy	Testing Accuracy
XGBoost	93.0377%	83.1993%
Random Forest	100%	78.6785%

KNN	83.6516%	73.3319%

Table 6:Testing and Training Accuracy for weed detection

The results of the weed detection implementation show that the XGBoost algorithm had the highest testing accuracy at 83.1993%. This means that the XGBoost algorithm was able to correctly identify weeds in the soybean crop images with a high degree of accuracy. The Random Forest algorithm had a training accuracy of 100%, which means that it was able to correctly identify all of the weeds in the training dataset. However, its testing accuracy was lower at 78.6785%, indicating that it may not perform as well when applied to new, unseen data. The KNN algorithm had the lowest testing accuracy at 73.3319%, indicating that it may not be the most effective option for this particular classification problem. Overall, these results suggest that the XGBoost algorithm may be the most effective option for accurately detecting weeds in soybean crop images.

4.6 Plant disease implementation

The Plant disease implementation is a system that uses a machine learning algorithm to predict types of disease affecting a plant from images. This module was implemented using a support vector machine (SVM) and a random forest algorithm. The code for the machine learning algorithms used are included in Appendix A.

The SVM has a training accuracy of 76.15% and a testing accuracy of 53.72%. This suggests that the model is overfitting to the training data, as its performance is much better on the training data compared to the testing data. Overfitting occurs when a model is too complex and learns the specific details and noise in the training data, which can cause the model to perform poorly on new, unseen data. The model was saved as 'plant_disease_model.h5' and subsequently used for the web page implementation. The results are summarized in the table below.

Algorithm name	Training Accuracy	Testing Accuracy
Support vector machine	76.15%	53.72%
Random Forest	99.995%.	57.28%.

Table 7: Accuracy result for plant disease implementation

4.7 The web interface implementation

The Agro-predict web interface was developed using the Streamlit library. It has a sidebar with options for navigating to various pages, including the homepage, a module for predicting rainfall, a module for recommending fertilizers, a module for recommending crops, and a module for detecting plant diseases and weeds.

In the rainfall prediction module, users can input temperature and humidity values, and the module will use a trained machine learning model (loaded from the file "Rainfall_prediction_trained_model.sav" to predict the expected rainfall for the specified values.

The fertilizer recommendation module allows users to input N, P, K, temperature, humidity, pH, and rainfall values, and the module will then use a trained machine learning model (loaded from the file "Fertilizer_recommendation_trained_model.sav" to recommend the most suitable fertilizer for the specified values.

The crop recommendation module enables users to input N, P, K, temperature, humidity, pH, and rainfall values, and the module will use a trained machine learning model (loaded from the file "Crop_recommendation_trained_model.sav" to predict the best crop for the given values. The module then converts the prediction number to the corresponding crop name and displays it to the user.

The plant disease prediction module allows users to upload an image of a plant and uses a trained machine learning model (loaded from the file "plant_disease_model2nd.h5") to predict the disease affecting the plant. The prediction is made by pre-processing the image, making a prediction using the model, and getting the class label with the highest probability. The class index is then converted to the corresponding disease name and displayed to the user.

The weed detection module allows the user to upload an image and receive a prediction for the type of weed present in the image. The prediction can be one of four labels: 'soybean', 'grass', 'soil', or 'broadleaf'. The module loads a machine-learning model and defines functions for pre-processing the image and making a prediction using the model.

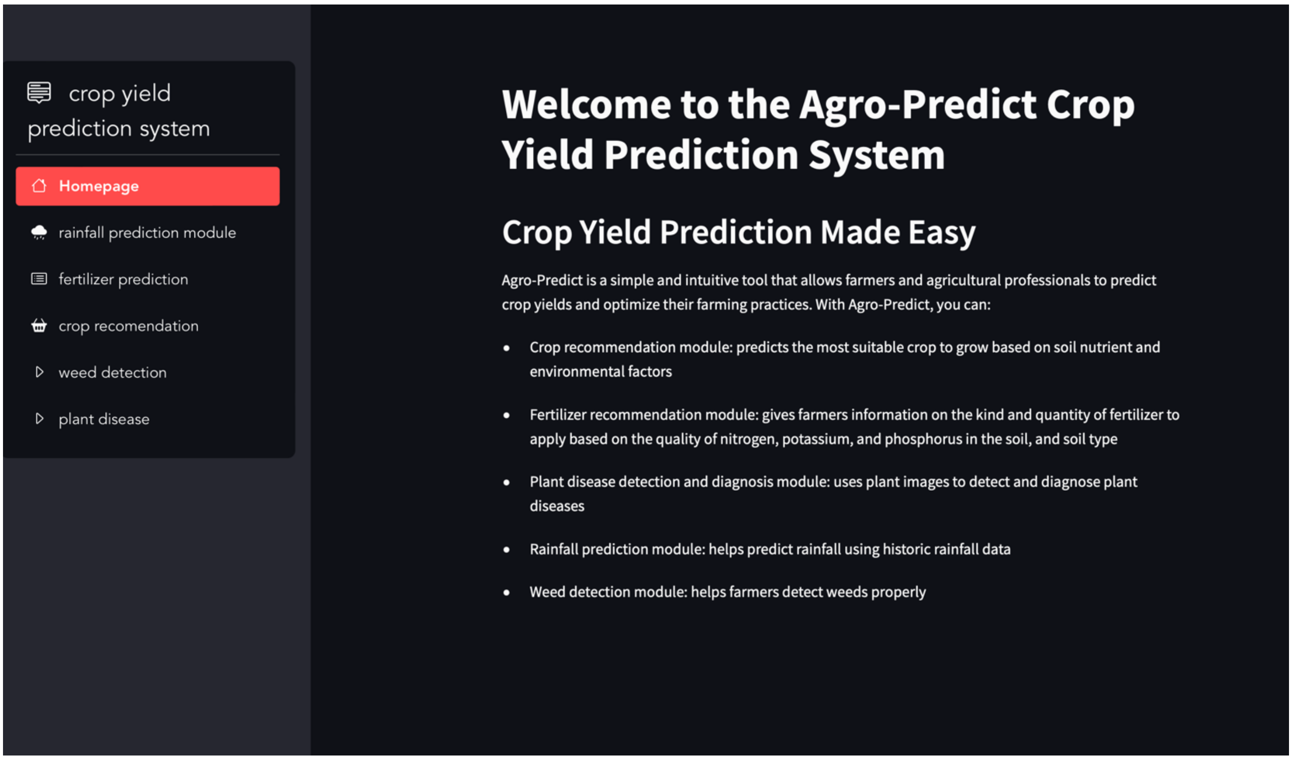

Figure 16: Agro-predict portal homepage

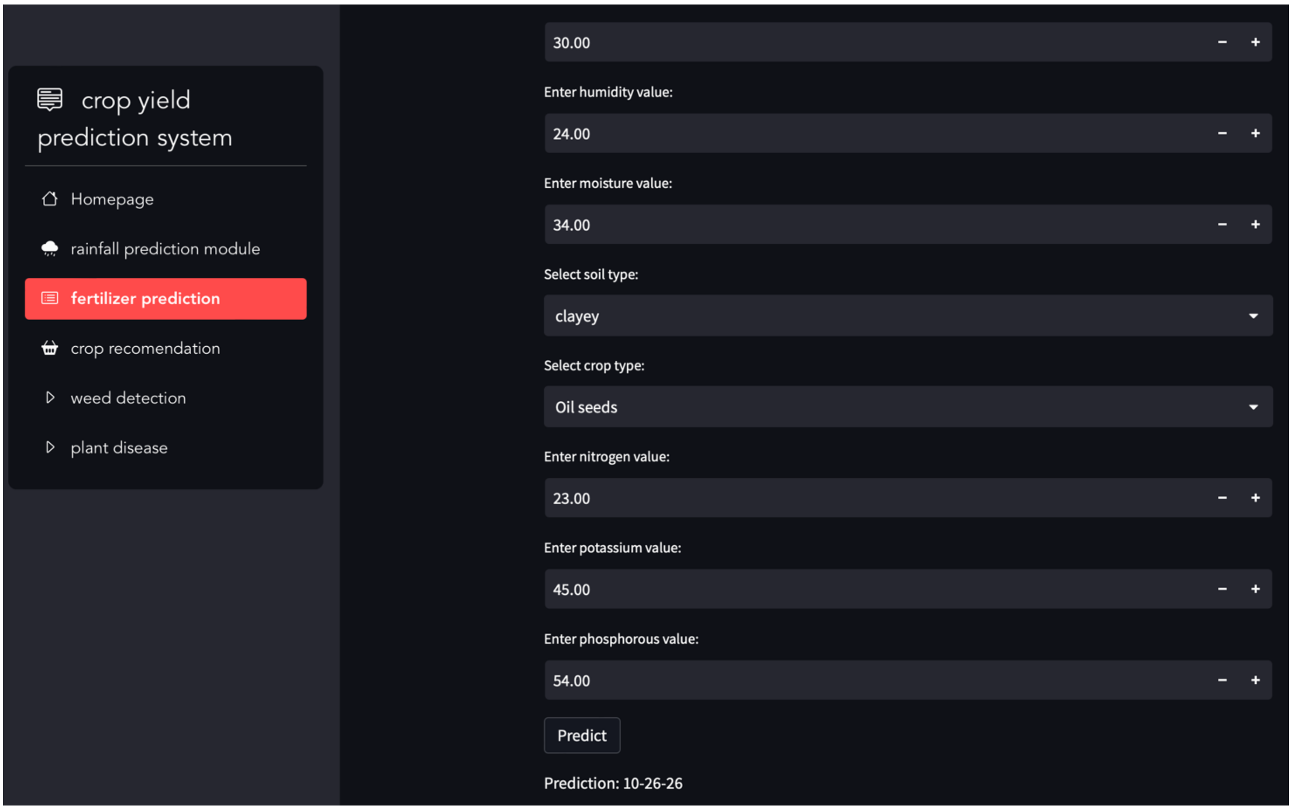

Figure 17: Fertilizer recommendation page

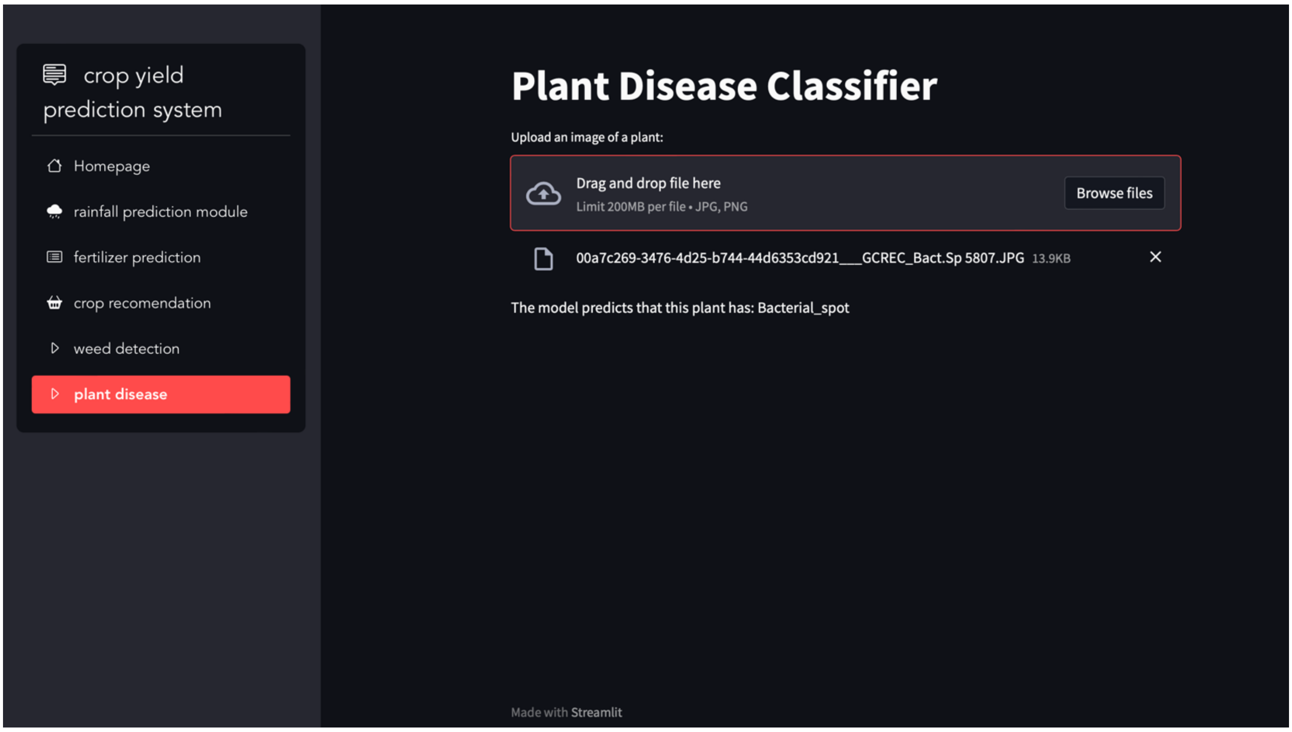

Figure 18:plant disease prediction page

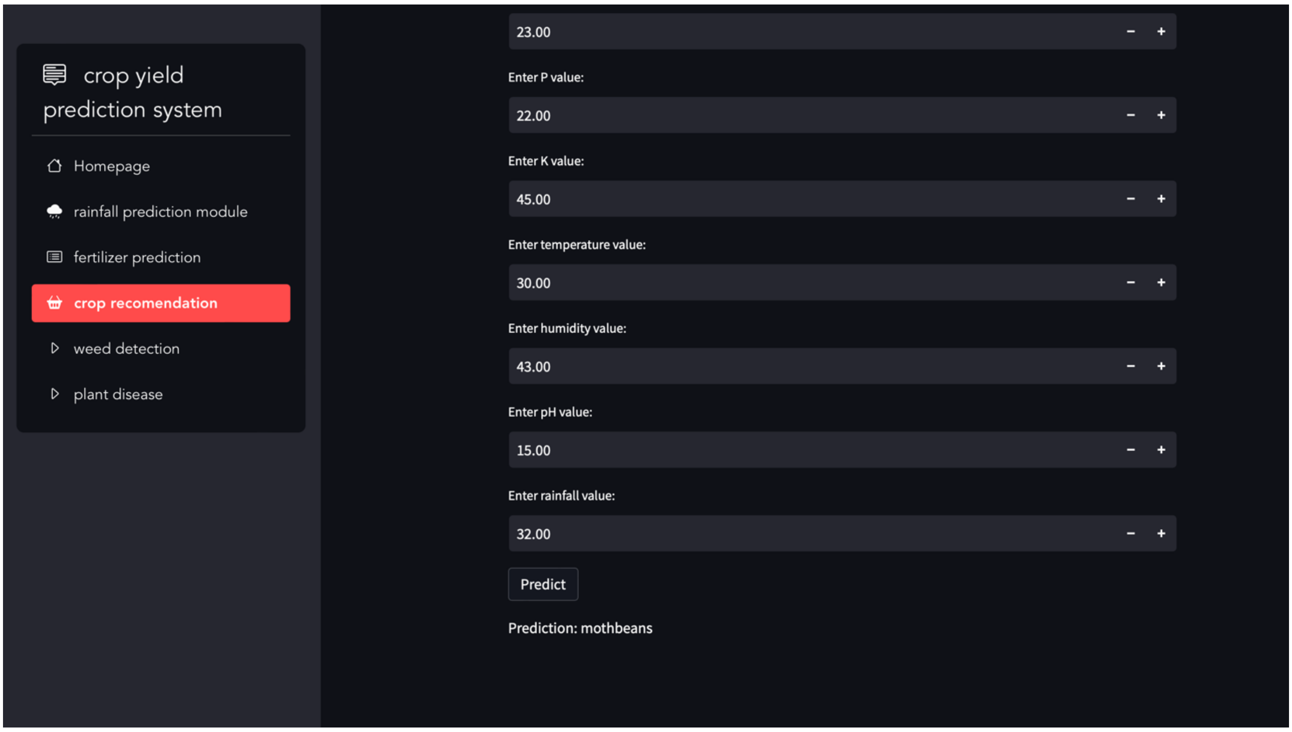

Figure 19:crop recommendation page

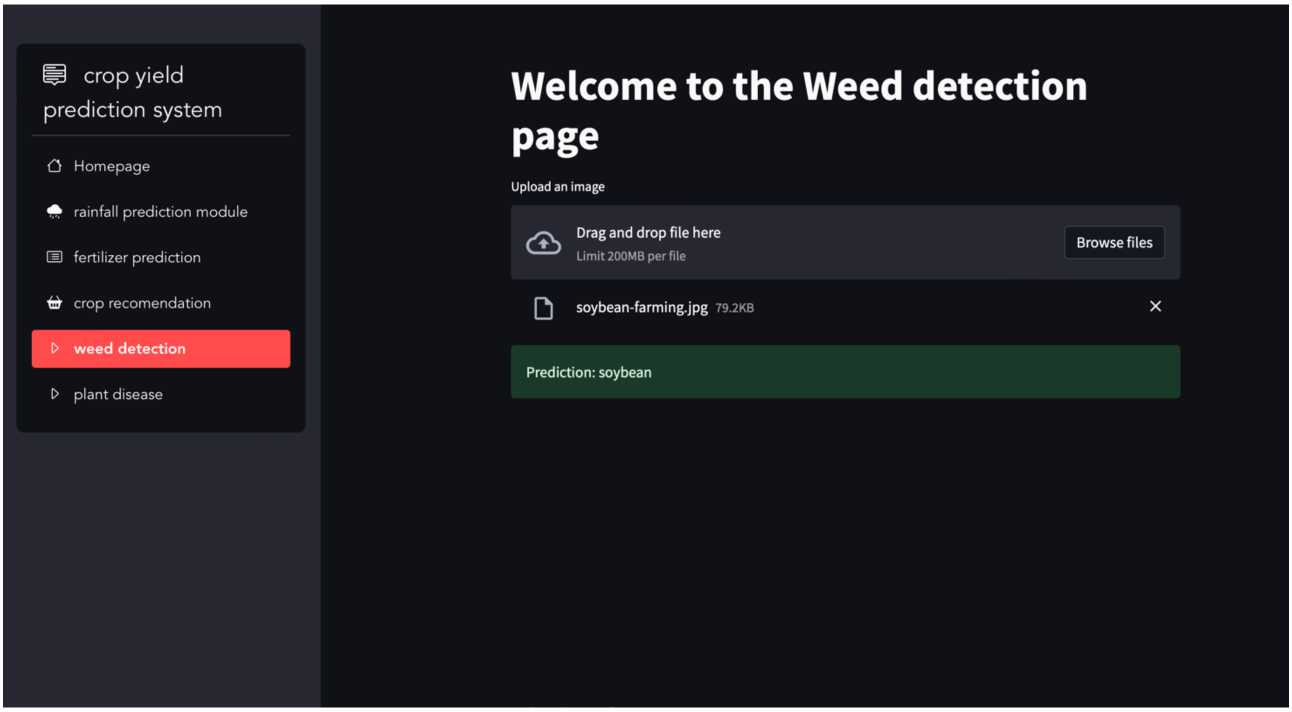

Figure 20:weed detection page

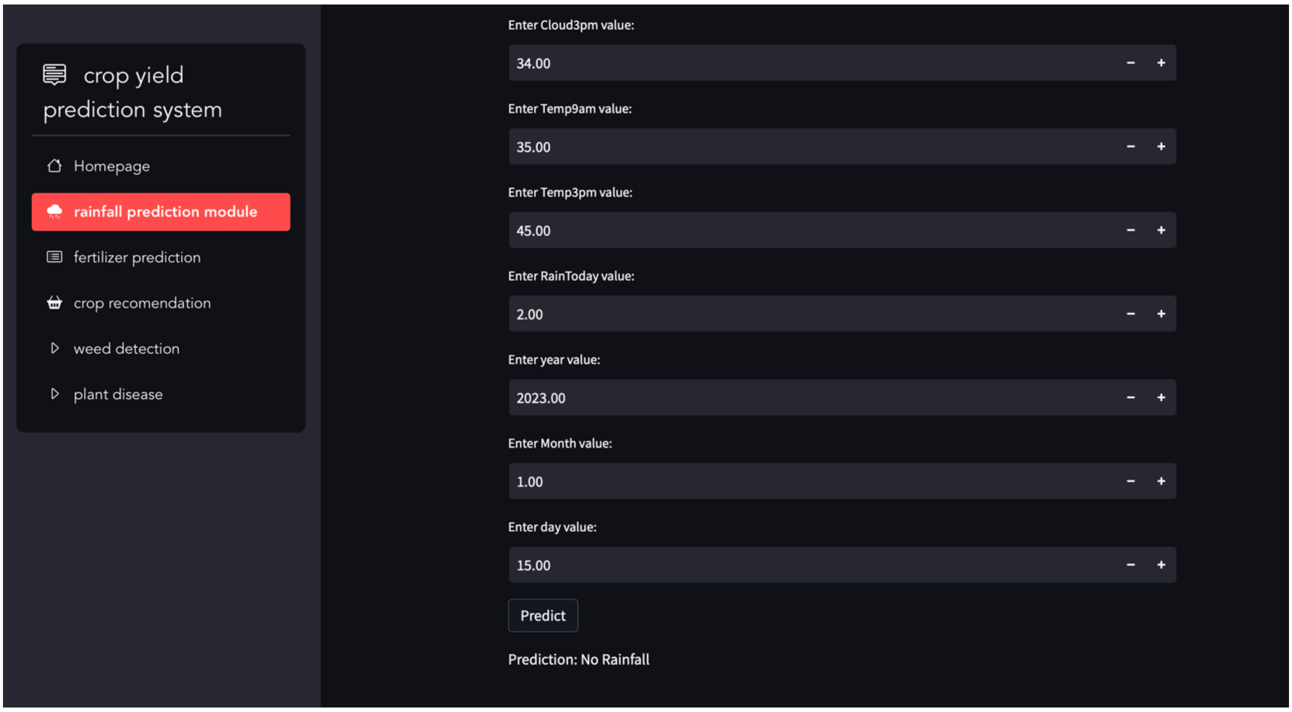

Figure 21:Rainfall prediction page

5 EVALUATION AND CONCLUSIONS

5.1 Critical Evaluation of aims and objectives

The following is a critical evaluation of the aims and objectives of the development of machine learning algorithms for improving crop yield prediction, The success of the project will be determined by its ability to effectively achieve the objectives outlined at the beginning of the project, which focus on the use of AI and machine learning technologies in improving agricultural practices and crop yield.

The project had three objectives

1. present an overview of machine learning applications in agriculture

This objective aimed to provide an overview of machine learning applications in agriculture and evaluate their potential for improving crop yield. To achieve this, I conducted a comprehensive review of the use of these technologies in various areas of agriculture, including an examination of their benefits and limitations. I also reviewed smart agriculture and climate-smart agricultural practices and recent studies on enhancing agricultural yield prediction.

Overall, this objective was successful in presenting a valuable and relevant understanding of the use of machine learning in agriculture. It demonstrated the impact of climate change on agriculture and the potential for climate-smart agriculture to address these challenges. The evaluation of smart agriculture practices contributed to an understanding of the potential of new technologies and approaches in improving agricultural practices. The review of recent studies on yield prediction also helped to identify the current state of research in this area and potential areas for further exploration.

2. Implement machine learning techniques to predict crop yield

This objective aimed to use machine learning techniques to predict crop yield by selecting and pre-processing data accessed from Kaggle and Mendeley. Four algorithms were trained on the acquired data and evaluated on test data using accuracy as the metric.

Overall, the objective of implementing machine learning techniques to predict crop yield was successful. The selected data were pre-processed and used to train the machine learning models, which were then evaluated on test data using accuracy as a metric. The success of this objective suggests that the machine learning techniques were able to effectively predict crop yield using the available data.

3. Implement a web-based interface for the predictive model developed to improve crop yield

Overall, the objective of implementing a web-based interface for a predictive model to improve crop yield was successful. The development of a user interface that allows farmers to input their own data and receive personalized predictions is a valuable tool for improving the effectiveness of the predictive model. In addition, the implementation of accuracy testing to evaluate the performance of the predictive model for all modules helped to ensure that the model was reliable and effective. Identifying the best-suited predictive model for all systems to improve crop yield contributed to the overall goal of improving crop yield through this tool. Overall, this objective achieved its aims and successfully contributed to the goal of improving crop yield through the use of a predictive model.

5.2 Conclusions

In conclusion, the development of machine learning algorithms for improving crop yield prediction is a significant step forward in addressing the challenges facing agriculture and supporting the economic and social development of communities around the world. By considering individual factors that contribute to crop yield, including weather patterns, soil conditions, pest and disease infestations, and crop recommendations, these algorithms can provide more accurate and comprehensive predictions that can help farmers to make informed decisions about their farming operations and improve crop yields. These algorithms demonstrated an accuracy ranging from 53% to 100%, with XGBoost and Random Forest achieving the highest accuracy for crop recommendation implementation, KNN achieving the highest testing accuracy for rainfall prediction, XGBoost achieving the highest accuracy for fertilizer prediction, XGBoost achieving the highest accuracy for weed detection, and Random Forest achieving the highest accuracy for plant disease prediction implementation.

5.3 Future work

There are several potential directions for future work that could build on the foundations established in this dissertation.

One potential area for future work based on this study is the implementation of these machine learning algorithms into a real data from fields on a farm or multiple farms, where they can be used to improve crop yield prediction and support decision-making for farmers. To achieve this, it may be necessary to conduct further testing and refinement of the algorithms, as well as integrate them into the existing systems and practices used on these farms.

Another area of focus could be exploring ways to make the prediction system more user-friendly and accessible for farmers. This could involve developing mobile apps or other tools such as SMS or MMS system that allows farmers to easily input data and receive predictions on the go, as well as exploring ways to provide personalized recommendations based on the specific conditions and needs of each farm.

A potential area for future work is exploring the potential for integration with other systems. For example, the algorithms developed in this study could be integrated with precision agriculture systems, which use sensors and other technologies to monitor and optimize farming operations. This integration could lead to more precise and targeted predictions and recommendations for farmers. In addition, integration with weather forecasting systems could allow for more accurate weather predictions, as well as real-time updates for farmers.

Another potential area for future work based on this study is the investigation of the potential economic and environmental impacts of implementing machine learning algorithms in agriculture. This could involve conducting cost-benefit analyses to determine the potential return on investment for farmers who adopt the algorithms, as well as evaluating the potential for reduced use of chemical inputs and other resources, such as water and fossil fuels, due to the more precise and targeted recommendations provided by the algorithms.

REFERENCES

Abate, T., van Huis, A., & Ampofo, J. K. O., 2000. "Pest management strategies in traditional agriculture: an African perspective." Annual Review of Entomology, 45(1), 631-659.
Abhishek, G.D. 2019. Fertilizer prediction [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/gdabhishek/fertilizer-prediction
About XGBoost. "About." Available at: https://xgboost.ai/about (Accessed: December 21, 2022).
Agriculture and climate change, 2021. "European Environment Agency." Available at: https://www.eea.europa.eu/signals/signals-2015/articles/agriculture-and-climate-change (Accessed: 22 Dec 2022).
Ahamed, A. M. S., Mahmood, N. T., Hossain, N., Kabir, M. T., Das, K., Rahman, F., & Rahman, R. M., 2015. "Applying data mining techniques to predict the annual yield of major crops and recommend planting different crops in different districts in Bangladesh." In: 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 1-6. IEEE.
Ahmed, Moiz Uddin, and Iqbal Hussain., 2022. "Prediction of Wheat Production Using Machine Learning Algorithms in northern areas of Pakistan." Telecommunications Policy 46, no. 6: 102370.
Ali, A. M., Abouelghar, M. A., Belal, A. A., Saleh, N., Younes, M., Selim, A., Emam, M. E. S., Elwesemy, A., Kucher, D. E., & Magignan, S., 2022. Crop yield prediction using multi sensors remote sensing. The Egyptian Journal of Remote Sensing and Space Science.
Almeyda, E., Paiva, J., & Ipanaqué, W., 2020. "Pest incidence prediction in organic banana crops with machine learning techniques." In: 2020 IEEE Engineering International Research Conference (EIRCON) (pp. 1-4). IEEE.
Andrée, P., 2005. "The Genetic Engineering Revolution in Agriculture and Food: Strategies of the." In: The business of global environmental governance, 135.
Aravatagimath, A., Sutagundar, A. V., & Yalavigi, V., 2021. Agriculture Product Marketing Data Analysis using Machine Learning. In 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS) (Vol. 1, pp. 1-6). IEEE.
Arsene, M. B., and Nkulu Mwine Fyama, 2021. "Potential threats to agricultural food production and farmers' coping strategies in the marshlands of Kabare in the Democratic Republic of Congo." Cogent Food & Agriculture, 7(1), 1933747.
Ashok, S., Kishore, G., Rajesh, V., Suchitra, S., Sophia, S. G., & Pavithra, B., 2020. "Tomato leaf disease detection using deep learning techniques." In: 2020 5th International Conference on Communication and Electronics Systems (ICCES) (pp. 979-983). IEEE.
Average temperature anomaly., 2022. "Our World in Data." Available at: https://ourworldindata.org/grapher/temperature-anomaly [Accessed 22 Dec 2022].
Bashir, A., Manzoor, T., Malik, L. A., Qureashi, A., & Pandith, A. H., 2020. "Enhanced and selective adsorption of Zn (II), Pb (II), Cd (II), and Hg (II) ions by a dumbbell-and flower-shaped

potato starch phosphate polymer: a combined experimental and DFT calculation study." ACS omega, 5(10), 4853-4867.
Beulah, R., 2019. "A survey on different data mining techniques for crop yield prediction." Int. J. Comput. Sci. Eng. 7(1), 738-744. https://doi.org/10.26438/ijcse/v7i1.738744
Bhanumathi, S., M. Vineeth, and N. Rohit, 2019. "Crop yield prediction and efficient use of fertilizers." In: International Conference on Communication and Signal Processing (ICCSP).
Boursianis, A.D., Papadopoulou, M.S., Diamantoulakis, P., Liopa-Tsakalidi, A., Barouchas, P., Salahas, G., Karagiannidis, G., Wan, S., and Goudos, S.K., 2022. "Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review." Internet of Things, 18, pp. 100187.
Brohm, K.A. and Klein, S., 2020. "THE CONCEPT OF CLIMATE SMART AGRICULTURE–A CLASSIFICATION IN SUSTAINABLE THEORIES." International Journal for Quality Research, 14(1).
Chakraborty, S. K., Chandel, N. S., Jat, D., Tiwari, M. K., Rajwade, Y. A., and Subeesh, A., 2022. Deep learning approaches and interventions for futuristic engineering in Agriculture. Neural Computing and Applications, 34 (23), 20539–20573.
Chlingaryan, A., Sukkarieh, S., & Whelan, B., 2018. "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review." Computers and electronics in agriculture, 151, 61-69.
Choudhary, S., Gaurav, V., Singh, A., & Agarwal, S., 2019. "Autonomous crop irrigation system using artificial intelligence." Int. J. Eng. Adv. Technol, 8(5), 46-51.
Dahikar, S. S., & Rode, S. V., 2014. Agricultural crop yield prediction using artificial neural
Dharmaraj, V., & Vijayanand, C., 2018. "Artificial intelligence (AI) in agriculture." International Journal of Current Microbiology and Applied Sciences, 7(12), 2122-2128.
Dhole, S., Mokashi, A., Kulkarni, M., & Shukla, S., 2022. "AgriBot:(Automation in farming with IOT)."
dos Santos Ferreira, A., Freitas, D.M., da Silva, G.G., Pistori, H. and Folhes, M.T., 2017. Weed detection in soybean crops using ConvNets. Computers and Electronics in Agriculture, 143, pp.314-324.
Durai, Senthil Kumar Swami, and Mary Divya Shamili, 2022 "Smart farming using Machine Learning and Deep Learning techniques." Decision Analytics Journal 3: 100041.
Durgabai, R. P. L., & Bhargavi, P., 2018. "Pest management using machine learning algorithms: a review." International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), 8(1), 13-22.
FAO (2022). Climate-smart agriculture case studies, 2021 – Projects from around the world. Rome: FAO. https://doi.org/10.4060/cb5359en
Fao.org, 2022. "Climate-Smart Agriculture | Food and Agriculture Organization of the United Nations." Available at: https://www.fao.org/climate-smart-agriculture/en/ (Accessed: 22 Dec 2022).

Farooq, MS, Riaz, S, Abid, A, Umer, T, Zikria, YB., 2020. "Role of IoT Technology in Agriculture: A Systematic Literature Review." Electronics. 9(2): 319. doi:10.3390/electronics9020319.

Ferentinos, K. P., 2018. "Deep learning models for plant disease detection and diagnosis." Computers and electronics in agriculture, 145, 311-318.

Food and Agriculture Organization of the United Nations (FAO), 2009. High Level Expert Forum—How to Feed the World in 2050.

Food and Agriculture Organization of the United Nations (FAO), 2009. High Level Expert Forum—How to Feed the World in 2050.

Geng, L., & Dong, T., 2017. An Agricultural Monitoring System Based on Wireless Sensor and Depth Learning Algorithm. Int. J. Online Eng., 13(12), 127-137.

Gollakota, Akhila, and M. B. Srinivas, 2011 "Agribot-A multipurpose agricultural robot, India Conference (INDICON).".

Ho, T. K., 1995. "Random decision forests." In: Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282). IEEE.

instrumentation and control engineering, 2(1), 683-686.

Islam, N., Rashid, M.M., Wibowo, S., Xu, C.Y., Morshed, A., Wasimi, S.A., Moore, S., and Rahman, S.M., 2021. "Early weed detection using image processing and machine learning techniques in an Australian chilli farm." Agriculture, 11(5), p.387.

Ivanyna, M., Stern, N., Oman, W., & Bhattacharya, A., 2021. "Climate Action to Unlock the Inclusive Growth Story of the 21st Century."

Jeong, S., Ko, J., & Yeom, J. M., 2022. "Predicting rice yield at pixel scale through synthetic use of crop and deep learning models with satellite data in South and North Korea." Science of the Total Environment, 802, 149726.

Jeong, S., Ko, J., & Yeom, J. M., 2022. Predicting rice yield at pixel scale through synthetic use of crop and deep learning models with satellite data in South and North Korea. Science of the Total Environment, 802, 149726.

Jha, K., Doshi, A., Patel, P., & Shah, M., 2019. "A comprehensive review on automation in agriculture using artificial intelligence." Artificial Intelligence in Agriculture, 2, 1-12.

Kale, S. S., & Patil, P. S., 2019. "A machine learning approach to predict crop yield and success rate." In: 2019 IEEE Pune Section International Conference (PuneCon)(pp. 1-5). IEEE

Khaki, S. and Wang, L., 2019. "Crop yield prediction using deep neural networks." Frontiers in plant science, 10, p.621.

Khaki, S. and Wang, L., 2019. Crop Yield Prediction Using Deep Neural Networks. Frontiers in Plant Science, 10

Khan Q., 2022. Tomato disease multiple sources [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/cookiefinder/tomato-disease-multiple-sources

Kim, N., Ha, K. J., Park, N. W., Cho, J., Hong, S., & Lee, Y. W., 2019. "A comparison between major artificial intelligence models for crop yield prediction: Case study of the midwestern United States, 2006–2015." ISPRS International Journal of Geo-Information, 8(5), 240.

Krishnan, A., & Swarna, S., 2020. "Robotics, IoT, and AI in the automation of agricultural industry: a review." In: 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC) (pp. 1-6). IEEE.

Kumar, A., Pooniwala, O. R., and Chakraborty, S., 2020. "Intelligent Crop Recommendation System Using ML." Doctoral dissertation, CMR Institute of Technology. Bangalore.

Kumar, M. S., Ganesh, D., Turukmane, A. V., Batta, U., & Sayyadliyakat, K. K., 2022. "Deep Convolution Neural Network Based solution for Detecting Plant Diseases." Journal of Pharmaceutical Negative Results, 464-471.

Kumar, Y. J. N., Spandana, V., Vaishnavi, V. S., Neha, K., & Devi, V. G. R. R., 2020. "Supervised machine learning approach for crop yield prediction in the agriculture sector." In: 2020 5th International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Lal, R., 2021. "Climate change and agriculture." In: Climate Change (pp. 661-686). Elsevier.

Liliane, T. N., & Charles, M. S., 2020. "Agronomy-Climate Change & Food Security." In: Factors Affecting Yield of Crops. IntechOpen.

Lipper, L., Thornton, P., Campbell, B.M., Baedeker, T., Braimoh, A., Bwalya, M., Caron, P., Cattaneo, A., Garrity, D., Henry, K. and Hottle, R., 2014. "Climate-smart agriculture for food security." Nature Climate Change, 4(12), pp.1068-1072.

Liu, B., & Bruch, R., 2020. "Weed detection for selective spraying: A review." Current Robotics Reports, 1(1), 19-26.

Lynch, L., and Brown, C., 2000. "Landowner decision making about riparian buffers." Journal of Agricultural and Applied Economics, 32(3), 585-596.

Marković, D., Vujičić, D., Tanasković, S., Đorđević, B., Ranđić, S., & Stamenković, Z., 2021. "Prediction of pest insect appearance using sensors and machine learning." Sensors, 21(14), 4846.

Medar, R., Rajpurohit, V. S., & Shweta, S., 2019. "Crop yield prediction using machine learning techniques." In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.

Menaka, K., and Yuvaraj, N., 2016. "A survey on crop yield prediction models." Indian Journal of Innovations and Developments, 5(12), 1-7.

Mendelsohn, R., 2009. "The impact of climate change on agriculture in developing countries." Journal of Natural Resources Policy Research, 1(1), 5-19.

Meshram, Vishal, Kailas Patil, Vidula Meshram, Dinesh Hanchate, and S. D. Ramkteke, 2021. "Machine learning in agriculture domain: A state-of-art survey." Artificial Intelligence in the Life Sciences 1: 100010.

Micle, D. E., Deiac, F., Olar, A., Drența, R. F., Florean, C., Coman, I. G., & Arion, F. H., 2021. Research on Innovative Business Plan. Smart Cattle Farming Using Artificial Intelligent Robotic Process Automation. Agriculture, 11(5

Mishra, S., Paygude, P., Chaudhary, S., & Idate, S., 2018. "Use of data mining in crop yield prediction." In: 2018 2nd International Conference on Inventive Systems and Control (ICISC), pp. 796-802. IEEE.

Morice, C., Kennedy, J., Rayner, N., and Jones, P., 2012. "Quantifying uncertainties in global and regional temperature change using an ensemble of observational estimates: The HadCRUT4 data set." Journal of Geophysical Research: Atmospheres, 117 (D8).

Nanushi, O., Sitokonstantinou, V., Tsoumas, I., & Kontoes, C., 2022. "Pest presence prediction using interpretable machine learning." arXiv preprint arXiv:2205.07723.

Nerger, M., 2022. "What is climate-smart agriculture?", Rainforest Alliance. Available at: https://www.rainforest-alliance.org/insights/what-is-climate-smart-agriculture/ (Accessed: November 24, 2022).

network approach. International journal of innovative research in electrical, electronics,

Nigam, A., Garg, S., Agrawal, A., & Agrawal, P., 2019. "Crop yield prediction using machine learning algorithms." In: 2019 Fifth International Conference on Image Information Processing (ICIIP), pp. 125-130. IEEE.

Nyasimi, M., Kimeli, P., Sayula, G., Radeny, M., Kinyangi, J., & Mungai, C., 2017. "Adoption and dissemination pathways for climate-smart agriculture technologies and practices for climate-resilient livelihoods in Lushoto, Northeast Tanzania." Climate, 5(3), 63.

Palombi, L. and Sessa, R., 2013. Climate-smart agriculture: sourcebook. Climate-smart agriculture: sourcebook.

Pant, J., Pant, R. P., Singh, M. K., Singh, D. P., & Pant, H., 2021. "Analysis of agricultural crop yield prediction using statistical techniques of machine learning." Materials Today: Proceedings, 46, 10922-10926.

Panwar, E., Kukunuri, A. N. J., Singh, D., Sharma, A. K., & Kumar, H., 2022. An Efficient Machine Learning Enabled Non-Destructive Technique for Remote Monitoring of Sugarcane Crop Health. IEEE Access, 10, 75956-75970.

Paudel, D., Boogaard, H., de Wit, A., van der Velde, M., Claverie, M., Nisini, L., Janssen, S., Osinga, S., and Athanasiadis, I.N., 2022. "Machine learning for regional crop yield forecasting in Europe." Field Crops Research, 276, p.108377.

Plant Disease: Pathogens and Cycles, 2022. "CropWatch." Available at: https://cropwatch.unl.edu/soybean-management/plant-disease [Accessed 22 November 2022].

Prasad, N. R., Patel, N. R., & Danodia, A., 2021. "Crop yield prediction in cotton for regional level using random forest approach." Spatial Information Research, 29(2), 195-206.

Precision Agriculture, 2022 Nesta. Available at: https://www.nesta.org.uk/feature/precision-agriculture/ (Accessed: November 20, 2022).

Ritchie, H., Roser, M., & Rosado, P., 2020. "CO_2 and greenhouse gas emissions." Our World in Data.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., ... & Bengio, Y., 2022. "Tackling climate change with machine learning." ACM Computing Surveys (CSUR), 55(2), 1-96.

Sajja, G. S., Jha, S. S., Mhamdi, H., Naved, M., Ray, S., & Phasinam, K., 2021. "An investigation on crop yield prediction using machine learning." In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 916-921). IEEE.

Saranaya, M. and Sathappan, D.S., 2020. “Multi-model ensemble depth adaptive deep neural network for crop yield prediction,” International Journal of Recent Technology and Engineering (IJRTE), 8(5), pp. 3098–3093. Available at: https://doi.org/10.35940/ijrte.e6337.018520.

Saya, L., Gautam, D., Malik, V., Singh, W. R., & Hooda, S., 2020. Natural polysaccharide based graphene oxide nanocomposites for removal of dyes from wastewater: a review. Journal of Chemical & Engineering Data, 66(1), 11-37.

Shah, V., & Shah, P., 2018. "Groundnut crop yield prediction using machine learning techniques." Int. J. Scient. Res. Comput. Sci. Eng. Inform. Technol, 3(5), 1093-1097.

Shah, V., & Shah, P., 2018. Groundnut crop yield prediction using machine learning techniques. Int. J. Scient. Res. Comput. Sci. Eng. Inform. Technol, 3(5), 1093-1097.

Sharma S., 2021. Crop recommendation dataset, Kaggle. Available at: https://www.kaggle.com/datasets/siddharthss/crop-recommendation-dataset (Accessed: January 6, 2023).

Skawsang, S., Nagai, M., K. Tripathi, N., & Soni, P., 2019. "Predicting rice pest population occurrence with satellite-derived crop phenology, ground meteorological observation, and machine learning: a case study for the Central Plain of Thailand." Applied Sciences, 9(22), 4846.

Smith, D. R., & White, D. G., 1988. Diseases of corn. Corn and corn improvement, 18, 687-766.

Snyder, C. S., Bruulsema, T. W., Jensen, T. L., and Fixen, P. E., 2009. "Review of greenhouse gas emissions from crop production systems and fertilizer management effects." Agriculture, Ecosystems & Environment, 133(3-4), 247-266.

Steenwerth, K.L., Hodson, A.K., Bloom, A.J., Carter, M.R., Cattaneo, A., Chartres, C.J., Hatfield, J.L., Henry, K., Hopmans, J.W., Horwath, W.R. and Jenkins, B.M., 2014. "Climate-smart agriculture global research agenda: scientific basis for action." Agriculture & Food Security, 3(1), pp.1-39.

Stocker, T.F., Qin, D., Plattner, G.K., Tignor, M.M., Allen, S.K., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P.M., 2014. "Climate Change 2013: The physical science basis. Contribution of working group I to the fifth assessment report of IPCC the intergovernmental panel on climate change."

Suma, N., Samson, S. R., Saranya, S., Shanmugapriya, G., and Subhashri, R., 2017. "IOT based smart agriculture monitoring system." International Journal on Recent and Innovation Trends in computing and communication, 5(2), 177-181.

Thilakarathne, N. N., Bakar, M. S. A., Abas, P. E., and Yassin, H., 2022. "A Cloud Enabled Crop Recommendation Platform for Machine Learning-Driven Precision Farming." Sensors, 22(16), 6299.

Tubiello, F.N., Salvatore, M., Cóndor Golec, R.D., Ferrara, A., Rossi, S., Biancalani, R., Federici, S., Jacobs, H., & Flammini, A., 2014. "Agriculture, forestry and other land use emissions by sources and removals by sinks." Rome, Italy.

United Nations Department of Economic and Social Affairs, Population Division, 2022. "World Population Prospects, 2022: Summary of Results." UN DESA/POP/2022/TR/NO. 3.

United Nations, 2022. "What Is Climate Change?" [online]. United Nations. Available from: https://www.un.org/en/climatechange/what-is-climate-change [Accessed 3 Jul 2022].

upstart., 2022. [online]. Available from: https://www.upstart.com/i/about [Accessed 27 Dec 2022].

US EPA, 2022. Sources of Greenhouse Gas Emissions | US EPA [online]. US EPA. Available from: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions#:~:text=Human%20activities%20are%20responsible%20for,over%20the%20last%20150%20years.&text=The%20largest%20source%20of%20greenhouse,electricity%2C%20heat%2C%20and%20transportation. [Accessed 27 Dec 2022].

Vadlamudi, S., 2019. "How Artificial Intelligence Improves Agricultural Productivity and Sustainability: A Global Thematic Analysis." Asia Pacific Journal of Energy and Environment, 6(2), 91-100.

Van Evert, F. K., Fountas, S., Jakovetic, D., Crnojevic, V., Travlos, I., Empenaar, C., 2017. "Disease Prediction of Big Data for weed control and crop protection", An International Journal of weed Biology, Ecology and Vegetation Management, Weed Research published by John Wiley & Sons Ltd on behalf of European Weed Research Society. 57, 218-233.

Van Klompenburg, Thomas, Ayalew Kassahun, and Cagatay Catal, 2020. "Crop yield prediction using machine learning: A systematic literature review." Computers and Electronics in Agriculture 177: 105709.

Veenadhari, S., Misra, B., & Singh, C. D., 2014. "Machine learning approach for forecasting crop yield based on climatic parameters." In: 2014 International Conference on Computer Communication and Informatics (pp. 1-5). IEEE.

Vinson Joshua, S., Selwin Mich Priyadharson, A., Kannadasan, R., Ahmad Khan, A., Lawanont, W., Ahmed Khan, F., Ur Rehman, A., and Junaid Ali, M., 2022. Crop yield prediction using machine learning approaches on a wide spectrum. Computers, Materials & Continua, 72 (3), 5663–5679.

Virk, A. L., Noor, M. A., Fiaz, S., Hussain, S., Hussain, H. A., Rehman, M., ... & Ma, W., 2020. "Smart farming: An overview." In: Smart Village Technology, pp. 191-201.

Wang, A., Zhang, W., & Wei, X., 2019. "A review on weed detection using ground-based machine vision and image processing techniques." Computers and electronics in agriculture, 158, 226-240.

World Bank, 2022. "Climate-smart agriculture" Available at: https://www.worldbank.org/en/topic/climate-smart-agriculture (Accessed: November 24, 2022).

Wu, Z., Chen, Y., Zhao, B., Kang, X., & Ding, Y., 2021. "Review of weed detection methods based on computer vision." Sensors, 21(11), 3647.

Young J., 2020. Rain in Australia [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package

APPENDIX A

Crop recommendation code.

```
#step1: Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder, StandardScaler
from sklearn.preprocessing import OneHotEncoder
#step 2: Importing the dataset
data =pd.read_csv("Crop_recommendation.csv")
#A look at the first ten row.
data.head(10).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
#step 3: Handling the missing Data
#checking the total missing value in each features
print('Total missing value :')
data.isna().sum()
#print Dataset information
print('\033[32m\033[1m'+' Dataset Info:')
print('\033[0m'+'Total Rows:'+'\033[36m\033[1m', data.shape[0])
print('\033[0m'+'Total Columns:'+'\033[36m\033[1m', data.shape[1])
data.info()
#checking dataset
data.isnull()
data.nunique()
data.dropna()
#step 4:Data description and features correlation
# describing dataset
data.describe()
#features correlation
cmap = sns.color_palette("Greens", as_cmap=True)
plt.figure(figsize=(10,8))
plt.title("Features Correlation", fontsize=24)
sns.heatmap(data.corr(), annot=True, cmap=cmap, cbar=False,vmin=-1, vmax=1)
plt.show()
```

```
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Nitrogen relation with crops", fontsize=24)
sns.barplot(x="label", y="N", data=data, palette="tab10", )
plt.xlabel("crops")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Phosphorus  relation with crops", fontsize=24)
sns.barplot(x="label", y="P", data=data, palette="tab10", )
plt.xlabel("crops")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Potassium  relation with crops", fontsize=24)
sns.barplot(x="label", y="K", data=data, palette="tab10", )
plt.xlabel("crops")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Temperature relation with crops", fontsize=24)
sns.barplot(x="label", y="temperature", data=data, palette="tab10")
plt.xlabel("crops")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("humidity relation with crops", fontsize=24)
sns.barplot(x="label", y="humidity", data=data, palette="tab10", )
plt.xlabel("crops")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("ph relation with crops", fontsize=24)
sns.barplot(x="label", y="ph", data=data, palette="tab10", )
plt.xlabel("crops")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("rainfall relation with crops", fontsize=24)
sns.barplot(x="label", y="rainfall", data=data, palette="tab10", )
plt.xlabel("crops")
 #dealing with the categorical data
categorical = data.select_dtypes(include=['object'])
categorical.head()
```

```
#label encoding of the categorical data
label_encode = LabelEncoder()
data['label'] = label_encode.fit_transform(data['label'])
crop_category = {index : label for index, label in enumerate(label_encode.classes_)}
crop_category
print(data['label'])
#Step 5: splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(data.drop('label', axis=1),
data['label'], test_size=0.25, random_state=42)
#printing the shape of the training set and test dataset
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
#printing the training set
print(X_train)
#printing the testing set
print(y_test)
X_train
X_test
y_train
y_test
#XGBoost Classification Algorithm
import xgboost
#XGBoost for training data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy for XGBoost: {:.4f}%".format(accuracy * 100))
#XGBoost for testing data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(X_train, y_train)
```

```
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy  for XGBoost: {:.4f}%".format(accuracy * 100))
#RandomForest Classifier
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
Training_accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy  for Random Forest: {:.4f}%".format(accuracy * 100))
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy for Random Forest: {:.4f}%".format(accuracy * 100))
#KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_train)
```

```
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_train, predictions)
# Print the accuracy
print("Training Accuracy for KNN: {:.4f}%".format(accuracy * 100))
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_test, predictions)

# Print the accuracy
print("Testing Accuracy for KNN: {:.4f}%".format(accuracy * 100))
best_accuracy = 0
best_model = None
if accuracy > best_accuracy:
   best_accuracy = accuracy
   best_model = model
print(model)
import joblib
joblib.dump(model, 'Crop_recommendation_trained_model.sav')
```

Fertilizer recommendation code.

```
#step1: Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder
#step 2: Importing the dataset
data =pd.read_csv("Fertilizer Prediction.csv")
##A look at the first ten row.
data.head(10).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
```

```
#step 3: Handling the missing Data
#checking the total missing value in each features
print('Total missing value :')
data.isna().sum()
#print Dataset information
print('\033[32m\033[1m'+' Dataset Info:')
print('\033[0m'+'Total Rows:'+'\033[36m\033[1m', data.shape[0])
print('\033[0m'+'Total Columns:'+'\033[36m\033[1m', data.shape[1])
data.info()
#step 4:Data description and features correlation
# describing dataset
data.describe()
data.nunique()
#features correlation
cmap = sns.color_palette("Greens", as_cmap=True)
plt.figure(figsize=(10,8))
plt.title("Features Correlation", fontsize=24)
sns.heatmap(data.corr(), annot=True, cmap=cmap, cbar=False,vmin=-1, vmax=1)
plt.show()
#Comparispon of Each Features with fertilizer  Name
plt.figure(figsize=(30,8))
plt.title("fertilizer relation with Temperature", fontsize=24)
sns.barplot(x="Fertilizer_Name", y="Temparature", data=data, palette="tab10", )
plt.xlabel("Fertilizer_Name")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Humidity   relation with Fertilizer_Name", fontsize=24)
sns.barplot(x="Fertilizer_Name", y="Humidity ", data=data, palette="tab10", )
plt.xlabel("Fertilizer_Name")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Moisture  relation with Fertilizer_Name", fontsize=24)
sns.barplot(x="Fertilizer_Name", y="Moisture", data=data, palette="tab10", )
plt.xlabel("Fertilizer_Name")
#Comparispon of Each Features with fertilizer  Name
plt.figure(figsize=(30,8))
plt.title("fertilizer relation with Nitrogen", fontsize=24)
sns.barplot(x="Fertilizer_Name", y="Nitrogen", data=data, palette="tab10", )
```

```
plt.xlabel("Fertilizer_Name")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Potassium  relation with Fertilizer_Name", fontsize=24)
sns.barplot(x="Fertilizer_Name", y="Potassium", data=data, palette="tab10", )
plt.xlabel("Fertilizer_Name")
#Comparispon of Each Features with Crops
plt.figure(figsize=(30,8))
plt.title("Phosphorous  relation with Fertilizer_Name", fontsize=24)
sns.barplot(x="Fertilizer_Name", y="Phosphorous", data=data, palette="tab10", )
plt.xlabel("Fertilizer_Name")
#Step 4: handling the missing Data
data.isnull()
data.notnull()
data.dropna()
#label encoding of the categorical data
encoder = OrdinalEncoder()
encode_cols = ['Soil_Type', 'Crop_Type', 'Fertilizer_Name']
encoder.fit(data[encode_cols])
data[encode_cols] = encoder.transform(data[encode_cols])
print(data)
#Step 5: splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(data.drop('Fertilizer_Name', axis=1),
data['Fertilizer_Name'], test_size=0.25, random_state=42)
#printing the shape of the training set and test dataset
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
#printing the training set
print(X_train)
#printing the testing set
print(X_test)
#XGBoost Classification Algorithm
print(data.tail(5))
import xgboost
#XGBoost for training data
# Fit the model to the train set
```

```
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy for XGBoost: {:.4f}%".format(accuracy * 100))
#XGBoost for testing data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy  for XGBoost: {:.4f}%".format(accuracy * 100))
#RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
Training_accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy  for Random Forest: {:.4f}%".format(accuracy * 100))
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
```

```
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy for Random Forest: {:.4f}%".format(accuracy * 100))
#KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_train)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_train, predictions)
# Print the accuracy
print("Training Accuracy for KNN: {:.4f}%".format(accuracy * 100))
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_test, predictions)
# Print the accuracy
print("Testing Accuracy for KNN: {:.4f}%".format(accuracy * 100))
best_accuracy = 0
best_model = None
if accuracy > best_accuracy:
   best_accuracy = accuracy
   best_model = model
print(model)
from joblib import dump, load
# Save the model to a file
dump(model, 'Fertilizer_recommendation_trained_model.sav')
```

Rainfall prediction code.

```
#step1: Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder
import warnings
warnings.filterwarnings("ignore")
#step 2: Importing the dataset
data =pd.read_csv("weatherAUS.csv")
##A look at the first ten row.
data.head(10).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
#step 3: Handling the missing Data
#checking the total missing value in each features
print('Total missing value :')
data.isna().sum()
#print Dataset information
print('\033[32m\033[1m'+' Dataset Info:')
print('\033[0m'+'Total Rows:'+'\033[36m\033[1m', data.shape[0])
print('\033[0m'+'Total Columns:'+'\033[36m\033[1m', data.shape[1])
data.info()
#removing null values
data = data.dropna()
print("The shape of the dataset after removing null values is : ", data.shape)

#decribing the data
data.describe().T

#print the pre-processed Dataset information
print('\033[32m\033[1m'+' Dataset Info:')
print('\033[0m'+'Total Rows:'+'\033[36m\033[1m', data.shape[0])
print('\033[0m'+'Total Columns:'+'\033[36m\033[1m', data.shape[1])
data.info()
#preprocessing the column data
```

```
data['Date'] = pd.to_datetime(data['Date'])
data['year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
data['day'] = data['Date'].dt.day
data.drop('Date',axis=1, inplace=True)
data[['year','Month', 'day']].head()
#dealing with the categorical data
categorical = data.select_dtypes(include=['object'])
categorical.head()
#label encoding of the categorical data
encoder = OrdinalEncoder()
encode_cols = ['Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm','RainToday','RainTomorrow']
encoder.fit(data[encode_cols])
data[encode_cols] = encoder.transform(data[encode_cols])
 #label encoding of the categorical data
from sklearn.preprocessing import LabelEncoder
label_encode = LabelEncoder()
data['Location'] = label_encode.fit_transform(data['Location'])
crop_category = {index : label for index, label in enumerate(label_encode.classes_)}
crop_category
print(data)
#checking the total missing value in each features
print('Total missing value :')
data.isna().sum()
#Step 4: Data description and features correlation
data.isnull()
##A look at the first ten row.
data.head(90).style.background_gradient(cmap='Greens').set_properties(**{'font-family': 'Segoe
UI'}).hide_index()
data.nunique()
#features correlation
cmap = sns.color_palette("Greens", as_cmap=True)
plt.figure(figsize=(20,15))
plt.title("Features Correlation", fontsize=24)
sns.heatmap(data.corr(), annot=True, cmap=cmap, cbar=False,vmin=-1, vmax=1)
plt.show()
#data visualisation of the features.
data.hist(figsize=(17,15), color="g");
```

```
data.notnull()
data.dropna()
#Step 5: splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(data.drop('RainTomorrow', axis=1),
data['RainTomorrow'], test_size=0.25, random_state=42)
#printing the shape of the training set and test dataset
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
#printing the training set
print(X_train)
#printing the testing set
print(y_test)
#XGBoost Classification Algorithm
import xgboost
#XGBoost for training data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy for XGBoost: {:.4f}%".format(accuracy * 100))
#XGBoost for testing data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy  for XGBoost: {:.4f}%".format(accuracy * 100))
#KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import accuracy_score
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_train)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_train, predictions)
# Print the accuracy
print("Training Accuracy for KNN: {:.4f}%".format(accuracy * 100))
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_test, predictions)
# Print the accuracy
print("Testing Accuracy for KNN: {:.4f}%".format(accuracy * 100))
#RandomForest Classifier
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
Training_accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy  for Random Forest: {:.4f}%".format(accuracy * 100))
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
```

```
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy for Random Forest: {:.4f}%".format(accuracy * 100))
best_accuracy = 0
best_model = None
if accuracy > best_accuracy:
    best_accuracy = accuracy
    best_model = model
print(model)
import joblib
joblib.dump(model, 'Rainfall_prediction_trained_model.sav')
```

Plant Disease prediction code.

```
 #step1: Importing the required libraries
import os
import glob
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import cv2
#displaying the file path
folder_paths = glob.glob('/Users/ismailaoshodi/Desktop/dissertation/Artefact/plant_disease/train/*')
folder_paths
 def show_images(num, folder_paths, func=None, color=None):
    plt.figure(figsize=(15, 18))
    for i in range(num):
        fp = np.random.choice(folder_paths, size=1, replace=False)
        label = fp[0].split('/')[-1]
        img = plt.imread(np.random.choice(glob.glob(fp[0]+'/*'), size=1, replace=False)[0])

        if func==None:
```

```
            plt.subplot(4, 4, i+1)
            plt.imshow(img, cmap=color)
            plt.title(label)
        else:
            fig, ax = plt.subplots(1, 2, figsize=(8, 8))
            ax = ax.ravel()
        plt.axis('off')
#previwing random images from the folder path
show_images(11, folder_paths)
#defining the training and testing path
train_path='/Users/ismailaoshodi/Desktop/dissertation/Artefact/plant_disease/train/'
test_path='/Users/ismailaoshodi/Desktop/dissertation/Artefact/plant_disease/valid/'
#previewing the training directory
classes = os.listdir(train_path)
print(f"Total classes : {len(classes)}")
for i in classes:
    print(f"Number of samples in {i} class: {len(os.listdir(train_path + '/' + i))}")
#preparaing the training path for visualisation
class_names = sorted(os.listdir(train_path))
n_classes   = len(class_names)
class_dis = [len(os.listdir(train_path + name)) for name in class_names]
# Visualization of the training class names
fig = px.pie(names=class_names, values=class_dis, hole=0.4,
             title="Training Class of the diseases",
             labels={'names':'class names', 'values':'class distribution'})
fig.show()
# Visualization of the training class names
fig = px.line(x=class_names, y=class_dis,)
fig.update_layout({'title':{'text':"Training Class of the diseases ","x":0.42}})
fig.show()
def load_images(path):
    folders = os.listdir(path)
    data = []
    labels = []
    for subdir in folders:
        for file in os.listdir(os.path.join(path, subdir)):
            img = cv2.imread(os.path.join(path, subdir, file))
            img = cv2.resize(img, (32, 32))
```

```
            img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            img = img.flatten()
            data.append(img)
            labels.append(subdir)
    return np.array(data), np.array(labels)
dataset_dir = '/Users/ismailaoshodi/Desktop/dissertation/Artefact/plant_disease'
X, y = load_images(os.path.join(dataset_dir, 'train'))
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the train set
predictions = model.predict(X_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
Training_accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy  for Random Forest: {:.4f}%".format(Training_accuracy * 100))
# Create a random forest classifier
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy for Random Forest: {:.4f}%".format(accuracy * 100))
#SVM classifier
# Create an SVM classifier with default parameters
model = svm.SVC()
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
```

```
predictions = model.predict(X_train)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_train, predictions)
# Print the accuracy
print("Training Accuracy for SVM: {:.4f}%".format(accuracy * 100))
# Create an SVM classifier with default parameters
model = svm.SVC()
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_test, predictions)
# Print the accuracy
print("Testing Accuracy for SVM: {:.4f}%".format(accuracy * 100))
from joblib import dump, load
# Save the classifier to an HDF5 file
dump(model, 'plant_disease_model.h5')
```

weed detection code.

```
 import numpy as np
import pandas as pd
import os
from PIL import Image
import matplotlib.pyplot as plt
import glob
#displaying the file path
folder_paths =
glob.glob('/Users/ismailaoshodi/Desktop/dissertation/Artefact/Data_for_Weed/dataset/*')
folder_paths
 def show_images(num, folder_paths, func=None, color=None):
    plt.figure(figsize=(15, 18))
    for i in range(num):
        fp = np.random.choice(folder_paths, size=1, replace=False)
        label = fp[0].split('/')[-1]
        img = plt.imread(np.random.choice(glob.glob(fp[0]+'/*'), size=1, replace=False)[0])
        if func==None:
            plt.subplot(4, 4, i+1)
```

```
        plt.imshow(img, cmap=color)
        plt.title(label)
      else:
        fig, ax = plt.subplots(1, 2, figsize=(8, 8))
        ax = ax.ravel()
      plt.axis('off')
 show_images(11, folder_paths)
# Load the data
def load_images(path):
  folders=os.listdir(path)
  data=[]
  label=[]
  for i in folders:
    images=os.listdir(path+'/'+i)
    for j in images:
      image=Image.open(path+'/'+i+'/'+j).convert('RGB')
      image=image.resize((32,32))
      arr=np.array(image)
      data.append(arr)
      label.append(i)
  t=pd.factorize(np.array(label))
  tgt=t[0]
  return np.array(data),tgt
x,y=load_images('/Users/ismailaoshodi/Desktop/dissertation/Artefact/Data_for_Weed/dataset')
# Reshape the data into 2-dimensional matrices
x = np.reshape(x, (x.shape[0], -1))
# Split the data into training and test sets
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=100)
#RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
# Create a random forest classifier with the same number of estimators as the XGBoost model
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(x_train, y_train)
# Make predictions on the train set
predictions = model.predict(x_train)
```

```
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
Training_accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy  for Random Forest: {:.4f}%".format(Training_accuracy * 100))
# Create a random forest classifier
model = RandomForestClassifier(n_estimators=100)
# Train the classifier using the training data
model.fit(x_train, y_train)
# Make predictions on the test set
predictions = model.predict(x_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy for Random Forest: {:.4f}%".format(accuracy * 100))
#KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(x_train, y_train)
# Make predictions on the test set
predictions = model.predict(x_train)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_train, predictions)
# Print the accuracy
print("Training Accuracy for KNN: {:.4f}%".format(accuracy * 100))
# Create a KNN classifier with k=3
model = KNeighborsClassifier(n_neighbors=3)
# Fit the model to the training data
model.fit(x_train, y_train)
# Make predictions on the test set
predictions = model.predict(x_test)
# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_test, predictions)
# Print the accuracy
print("Testing Accuracy for KNN: {:.4f}%".format(accuracy * 100))
#XGBoost Classification Algorithm
```

```
import xgboost
#XGBoost for training data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(x_train, y_train)
# Make predictions on the train set
predictions = model.predict(x_train)
# Evaluate the model's performance on the train set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_train, predictions)
print("Training Accuracy for XGBoost: {:.4f}%".format(accuracy * 100))
#XGBoost for testing data
# Fit the model to the train set
model = xgboost.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=100)
model.fit(x_train, y_train)
# Make predictions on the test set
predictions = model.predict(x_test)
# Evaluate the model's performance on the test set
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predictions)
print("Testing Accuracy  for XGBoost: {:.4f}%".format(accuracy * 100))
from joblib import dump, load
# Save the model to a file
dump(model, 'weed_detection_model.sav')
```

The web interface code.

```
#import the model
import joblib
import streamlit as st
from streamlit_option_menu import option_menu
import numpy as np
from PIL import Image
from skimage.io import imread
from skimage.transform import resize
import tensorflow as tf
# Load the saved models from a file
```

```
rainfall_model= joblib.load("/Users/ismailaoshodi/Desktop/Rainfall_prediction_trained_model.sav")
crop_model=joblib.load("/Users/ismailaoshodi/Desktop/Crop_recommendation_trained_model.sav"
)
fertilizer_model=joblib.load("/Users/ismailaoshodi/Desktop/Fertilizer_recommendation_trained_mo
del.sav")
final_plant_model = tf.keras.models.load_model("plant_disease_model.h5")
weed_model=joblib.load("/Users/ismailaoshodi/Desktop/dissertation/Artefact/weed_detection_mod
el.sav" )
#side bar for navigate
with st.sidebar:
    selected=option_menu('crop yield prediction system',
                    ['Homepage',
                     'rainfall prediction module',
                    'fertilizer prediction',
                    'crop recomendation',
                    'weed detection',
                    'plant disease']
                    icons=['house','cloud-drizzle-fill','card-list', 'basket2-fill' ],
                    default_index=0)
if selected == 'Homepage':
    # Page title
    st.title('Welcome to the Agro-Predict Crop Yield Prediction System')
    # Page header
    st.header('Crop Yield Prediction Made Easy')
    # Introduction
    st.write('Agro-Predict is a simple and intuitive tool that allows farmers and agricultural
professionals to predict crop yields and optimize their farming practices. With Agro-Predict, you
can:')
    # Unordered list
    st.write('- Crop recommendation module: predicts the most suitable crop to grow based on soil
nutrient and environmental factors')
    st.write('- Fertilizer recommendation module: gives farmers information on the kind and quantity
of fertilizer to apply based on the quality of nitrogen, potassium, and phosphorus in the soil, and
soil type')
    st.write('- Plant disease detection and diagnosis module: uses plant images to detect and
diagnose plant diseases')
    st.write('- Rainfall prediction module: helps predict rainfall using historic rainfall data')
    st.write('- Weed detection module: helps farmers detect weeds properly')
```

```
    #crop recomendation page
if(selected  == 'crop recomendation'):
    #page title
    st.title('welcome to the crop recommendation  module ')
        # Collect input from the user
    N = st.number_input('Enter N value: ')
    P = st.number_input('Enter P value: ')
    K = st.number_input('Enter K value: ')
    temperature = st.number_input('Enter temperature value: ')
    humidity = st.number_input('Enter humidity value: ')
    ph = st.number_input('Enter pH value: ')
    rainfall = st.number_input('Enter rainfall value: ')
    # Add a predict button
    if st.button('Predict'):
        # Run the input through the model
        input_data = [[N, P, K, temperature, humidity, ph, rainfall]]
        prediction = crop_model.predict(input_data)
        # Convert the prediction number to the corresponding crop name
        if prediction[0] == 0:
            crop_name = 'apple'
        elif prediction[0] == 1:
            crop_name = 'banana'
        elif prediction[0] == 2:
            crop_name = 'blackgram'
        elif prediction[0] == 3:
            crop_name = 'chickpea'
        elif prediction[0] == 4:
            crop_name = 'coconut'
        elif prediction[0] == 5:
            crop_name = 'coffee'
        elif prediction[0] == 6:
            crop_name = 'cotton'
        elif prediction[0] == 7:
            crop_name = 'grapes'
        elif prediction[0] == 8:
            crop_name = 'jute'
        elif prediction[0] == 9:
```

```
            crop_name = 'kidneybeans'
        elif prediction[0] == 10:
            crop_name = 'lentil'
        elif prediction[0] == 11:
            crop_name = 'maize'
        elif prediction[0] == 12:
            crop_name = 'mango'
        elif prediction[0] == 13:
            crop_name = 'mothbeans'
        elif prediction[0] == 14:
            crop_name = 'mungbean'
        elif prediction[0] == 15:
            crop_name = 'muskmelon'
        elif prediction[0] == 16:
            crop_name = 'orange'
        elif prediction[0] == 17:
            crop_name = 'papaya'
        elif prediction[0] == 18:
            crop_name = 'pigeonpeas'
        elif prediction[0] == 19:
            crop_name = 'pomegranate'
        elif prediction[0] == 20:
            crop_name = 'rice'
        elif prediction[0] == 21:
            crop_name = 'watermelon'
        st.write(f'Prediction: {crop_name}')
    #fertilizer prediction page
if(selected  == 'fertilizer prediction'):
    #page title
    st.title('welcome to fertilizer prediction module using ml')
        # Collect input from the user
    temperature = st.number_input('Enter temperature value: ')
    humidity = st.number_input('Enter humidity value: ')
    moisture = st.number_input('Enter moisture value: ')
    soil_type = st.selectbox('Select soil type:', ['black', 'clayey', 'Loamy', 'Red','sandy' ])
    if soil_type == 'black':
        soil_type = 0
    elif soil_type == 'clayey':
```

```
        soil_type = 1
    elif soil_type == 'Loamy':
        soil_type= 2
    elif soil_type == 'Red':
        soil_type= 3
    elif soil_type == 'sandy':
        soil_type= 4
    crop_type = st.selectbox('Select crop type:', ['Barley', 'Cotton', 'Ground Nuts', 'Maize', 'Millets',
'Oil seeds', 'Paddy', 'Pulses', 'Sugarcane', 'Tobacco', 'Wheat'])
    # Convert crop_type to a numerical value
    if crop_type == 'Barley':
        crop_type = 0
    elif crop_type == 'Cotton':
        crop_type = 1
    elif crop_type == 'Ground Nuts':
        crop_type = 2
    elif crop_type == 'Maize':
        crop_type = 3
    elif crop_type == 'Millets':
        crop_type = 4
    elif crop_type == 'Oil seeds':
        crop_type = 5
    elif crop_type == 'Paddy':
        crop_type = 6
    elif crop_type == 'Pulses':
        crop_type = 7
    elif crop_type == 'Sugarcane':
        crop_type = 8
    elif crop_type == 'Tobacco':
        crop_type = 9
    elif crop_type == 'Wheat':
        crop_type = 10
    nitrogen = st.number_input('Enter nitrogen value: ')
    potassium = st.number_input('Enter potassium value: ')
    phosphorous = st.number_input('Enter phosphorous value: ')
    # Add a predict button
    if st.button('Predict'):
        # Run the input through the model
```

```
        input_data = [[temperature, humidity, moisture, soil_type, crop_type, nitrogen, potassium,
phosphorous]]
        prediction = fertilizer_model.predict(input_data)
        # Convert the prediction number to the corresponding fertilizer name
        if prediction[0] == 0:
            fertilizer_name = '10-26-26'
        elif prediction[0] == 1:
            fertilizer_name = '14-35-14'
        elif prediction[0] == 2:
            fertilizer_name = '17-17-17'
        elif prediction[0] == 3:
            fertilizer_name = '20-20'
        elif prediction[0] == 4:
            fertilizer_name = '28-28'
        elif prediction[0] == 5:
            fertilizer_name = 'DAP'
        elif prediction[0] == 6:
            fertilizer_name = 'Urea'
        st.write(f'Prediction: {fertilizer_name}')
    #rainfall prediction module
if(selected  == 'rainfall prediction module'):
    #page title
    st.title('welcome to rainfall prediction  module using ml')
    # Collect input from the user
    Location = st.selectbox('Select Location:', ['Adelaide', 'Albany', 'Albury', 'Alice Springs', 'Badgerys
Creek', 'Ballarat', 'Bendigo', 'Brisbane', 'Cairns', 'Canberra', 'Cobar', 'Coffs Harbour', 'Dartmoor',
'Darwin', 'Gold Coast', 'Hobart', 'Katherine', 'Launceston', 'Melbourne', 'Melbourne Airport', 'Mildura',
'Moree', 'Mount Gambier', 'Mount Ginini', 'Newcastle', 'Nhil', 'Norah Head', 'Norfolk Island',
'Nuriootpa', 'Pearce RAAF', 'Penrith', 'Perth', 'Perth Airport', 'Portland', 'Richmond', 'Sale', 'Salmon
Gums', 'Sydney', 'Sydney Airport', 'Townsville', 'Tuggeranong', 'Uluru', 'Wagga Wagga', 'Walpole',
'Watsonia', 'Williamtown', 'Witchcliffe', 'Wollongong', 'Woomera'])
    # Convert Location to a numerical value
    if Location == 'Adelaide':
        Location = 0
    elif Location == 'Albany':
        Location = 1
    elif Location == 'Albury':
        Location = 2
```

```
elif Location == 'Alice Springs':
    Location = 3
elif Location == 'Badgerys Creek':
    Location = 4
elif Location == 'Ballarat':
    Location = 5
elif Location == 'Bendigo':
    Location = 6
elif Location == 'Brisbane':
    Location = 7
elif Location == 'Cairns':
    Location = 8
elif Location == 'Canberra':
    Location = 9
elif Location == 'Cobar':
    Location = 10
elif Location == 'Coffs Harbour':
    Location = 11
elif Location == 'Dartmoor':
    Location = 12
elif Location == 'Darwin':
    Location = 13
elif Location == 'Gold Coast':
    Location = 14
elif Location == 'Hobart':
    Location = 15
elif Location == 'Katherine':
    Location = 16
elif Location == 'Launceston':
    Location = 17
elif Location == 'Melbourne':
    Location = 18
elif Location == 'Melbourne Airport':
    Location = 19
elif Location == 'Mildura':
    Location = 20
elif Location == 'Moree':
    Location = 21
```

```
elif Location == 'Mount Gambier':
    Location = 22
elif Location == 'Mount Ginini':
    Location = 23
elif Location == 'Newcastle':
    Location = 24
elif Location == 'Nhil':
    Location = 25
elif Location == 'Norah Head':
    Location = 26
elif Location == 'Norfolk Island':
    Location = 27
elif Location == 'Nuriootpa':
    Location = 28
elif Location == 'Pearce RAAF':
    Location = 29
elif Location == 'Penrith':
    Location = 30
elif Location == 'Perth':
    Location = 31
elif Location == 'Perth Airport':
    Location=32
elif Location == 'Portland':
    Location = 33
elif Location == 'Richmond':
    Location = 34
elif Location == 'Sale':
    Location = 35
elif Location == 'SalmonGums':
    Location = 36
elif Location == 'Sydney':
    Location = 37
elif Location == 'Sydney Airport':
    Location = 38
elif Location == 'Townsville':
    Location = 39
elif Location == 'Tuggeranong':
    Location = 40
```

```
elif Location == 'Uluru':
   Location = 41
elif Location == 'Wagga Wagga':
   Location = 42
elif Location == 'Walpole':
   Location = 43
elif Location == 'Watsonia':
   Location = 44
elif Location == 'Williamtown':
   Location = 45
elif Location == 'Witchcliffe':
   Location = 46
elif Location == 'Wollongong':
   Location = 47
elif Location == 'Woomera':
   Location = 48
MinTemp = st.number_input('Enter MinTemp value: ')
MaxTemp = st.number_input('Enter MaxTemp value: ')
Rainfall = st.number_input('Enter Rainfall value: ')
Evaporation = st.number_input('Enter Evaporation value: ')
Sunshine = st.number_input('Enter Sunshine value: ')
WindGustDir = st.number_input('Enter WindGustDir: ')
WindGustSpeed = st.number_input('Enter WindGustSpeed value: ')
WindDir9am = st.number_input('Enter WindDir9am: ')
WindDir3pm = st.number_input('Enter WindDir3pm: ')
WindSpeed9am = st.number_input('Enter WindSpeed9am value: ')
WindSpeed3pm = st.number_input('Enter WindSpeed3pm value: ')
Humidity9am = st.number_input('Enter Humidity9am value: ')
Humidity3pm = st.number_input('Enter Humidity3pm value: ')
Pressure9am = st.number_input('Enter Pressure9am value: ')
Pressure3pm = st.number_input('Enter Pressure3pm value: ')
Cloud9am = st.number_input('Enter Cloud9am value: ')
Cloud3pm = st.number_input('Enter Cloud3pm value: ')
Temp9am = st.number_input('Enter Temp9am value: ')
Temp3pm = st.number_input('Enter Temp3pm value: ')
RainToday = st.number_input('Enter RainToday value: ')
year = st.number_input('Enter year value: ')
Month = st.number_input('Enter Month value: ')
```

```
    day = st.number_input('Enter day value: ')
    # Add a predict button
    if st.button('Predict'):
        # Run the input through the model
        input_data = [[Location, MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustDir,
WindGustSpeed, WindDir9am, WindDir3pm, WindSpeed9am, WindSpeed3pm, Humidity9am,
Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm,
RainToday, year, Month, day]]
        prediction = rainfall_model.predict(input_data)
        # Convert the prediction number to the corresponding rainfall tomorrow value
        if prediction == 0:
            rain_tomorrow = 'No Rainfall '
        elif prediction == 1:
            rain_tomorrow = 'Possible Rainfall'
            st.write(f'Prediction: {rain_tomorrow}')
#plant disease prediction module
if selected == 'plant disease':
   # Preprocessing function
    def preprocess_image(image_path):
      im = imread(image_path)
      im = resize(im, (224,224,3))
      return im
    # Load the saved model
    # Set up the main app
    st.title("Plant Disease Classifier")
    # Get the image file from the user
    image_file = st.file_uploader("Upload an image of a plant:", type=["jpg", "png"])
    if image_file is not None:
      # Preprocess the image
      image = preprocess_image(image_file)
      # Make a prediction using the model
      prediction = final_plant_model.predict(np.expand_dims(image, axis=0))
      # Get the class label with the highest probability
      class_idx = np.argmax(prediction[0])
      # Convert the class index to the class name
      disease_names = ["Late_blight", "Septoria_leaf_spot",'Tomato_Yellow_Leaf_Curl_Virus',
'powdery_mildew', 'healthy', 'Early_blight', 'Target_Spot', 'Leaf_Mold','Tomato_mosaic_virus',
'Bacterial_spot', 'Spider_mites Two-spotted_spider_mite' ]
```

```
      class_name = disease_names[class_idx]

      # Display the prediction
      st.write(f"The model predicts that this plant has: {class_name}")
if(selected=='weed detection'):
        #page title
        st.title('Welcome to the Weed detection page ')
    # Load the model
weed_model=joblib.load("/Users/ismailaoshodi/Desktop/dissertation/Artefact/weed_detection_mod
el.sav" )
    # Preprocessing function
        def preprocess_image(image):
          # Resize the image to (32, 32)
          image = image.resize((32, 32))
          # Convert the image to an array
          image_array = np.array(image)
          # Reshape the image to a 2D matrix
          image_array = np.reshape(image_array, (1, -1))
          return image_array
    # Make a prediction function
        def make_prediction(image_array):
      # Make a prediction using the model
          prediction = weed_model.predict(image_array)
      # Return the prediction
          return prediction
    # Get the file path from the user
        file_path = st.file_uploader('Upload an image')
        if file_path is not None:
      # Load the image
          image = Image.open(file_path)
      # Preprocess the image
          image_array = preprocess_image(image)
      # Make a prediction
          prediction = make_prediction(image_array)
      # Dictionary to map prediction values to labels
          prediction_labels = {
          0: 'soybean',
          1: 'grass',
```

```
    2: 'soil',
    3: 'broadleaf'
}
# Get the label for the prediction
    prediction_label = prediction_labels[prediction[0]]
# Display the prediction
    st.success(f'Prediction: {prediction_label}')
```

APPENDIX B

- List of the contents of the Large File (code/artefact) submission on Brightspace

1. Crop recommendation.ipynb
2. Fertilizer recommendation.ipynb
3. Main.py
4. Plant disease prediction.ipynb
5. Rainfall prediction.ipynb
6. Weed_detection.ipynb
7. Crop_recommendation_trained_model.sav
8. Fertilizer_recommendation_trained_model.sav
9. plant_disease_model.h5
10. Rainfall_prediction_trained_model.sav
11. weed_detection_model.sav
12. video presentation

APPENDIX C

Bournemouth University

Research Ethics Checklist

About Your Checklist	
Ethics ID	44091
Date Created	22/04/2022 05:50:49
Status	Approved
Date Approved	22/04/2022 09:35:39
Date Submitted	22/04/2022 06:02:37
Risk	Low

Researcher Details	
Name	Ismaila Oshodi
Faculty	Faculty of Science & Technology
Status	Postgraduate Taught (Masters, MA, MSc, MBA, LLM)
Course	MSc Data Science & Artificial Intelligence

Project Details	
Title	Development of machine Learning Algorithms for improving crop yield prediction.
Start Date of Project	16/05/2022
End Date of Project	14/09/2022
Proposed Start Date of Data Collection	16/05/2022
Supervisor	Avleen Malhi
Approver	Avleen Malhi
Summary - no more than 600 words (including detail on background methodology, sample, outcomes, etc.)	

The proposed project will contain the development of five systems that contribute to improving crop yield. firstly, a construction of a machine learning system that predicts the best type of crops to be grown on soil based on the quantity of nitrogen, phosphorous, potassium and other factors present in the soil. Secondly a construction of machine learning system that predicts plant disease on farmland. The third system is a weather prediction system, it gives information about precipitation and other weather conditions in the area. The fourth system will be used for pest control and management that avoid total reliance on pesticides, it uses image processing detect pest on farmland, The fifth system is a weed management system, it is used in early detection of weeds on farmland. The development of these systems would help farmers improves his crop yield. The Proposed project will also contain a comprehensive review of how machine learning is being used in agriculture, a systematic literature review of climate-smart agriculture practises and a review of precision agriculture as well as their benefit and limitations.

Filter Question: Does your study involve the use or re-use of data which will be obtained from a source other than directly from a Research Participant?

Additional Details	
Please describe the data, its source and how you are permitted to use it	licensed data will be sourced from various online data repositories such as Kaggle.

Research Data	
Will identifiable personal information be collected, i.e. at an individualised level in a form that identifies or could enable identification of the participant?	No
Will research outputs include any identifiable personal information i.e. data at an individualised level in a form which identifies or could enable identification of the individual?	No

Storage, Access and Disposal of Research Data	
Where will your research data be stored and who will have access during and after the study has finished.	
The research data will be available online in the data repository and BU's Online research data repository(BORDaR) after the completi of the project.	
Once your project completes, will any anonymised research data be stored on BU's Online Research Data Repository "BORDaR"?	Yes

Final Review	
Are there any other ethical considerations relating to your project which have not been covered above?	No

Risk Assessment	
Have you undertaken an appropriate Risk Assessment?	Yes

APPENDIX D

Project Proposal Form
Please refer to the **Project Handbook** when completing this form and note that your proposal should be your own original work and you must cite sources in line with university guidance on **referencing and plagiarism**[1].

Student's Name:	Ismaila Oshodi
Degree Title:	MSC Data science and artificial intelligence
Project Title/Area:	Development of machine Learning Algorithms for improving crop yield prediction.
Supervisor's Name:	Dr. Avleen Malhi

Section 1: Project Overview

1.1 Problem definition - use one sentence to summarise the problem:

Crop yield prediction is an important agricultural problem, farmers have always been faced with the problem of not knowing the condition of the weather for the planting season, early detection of plant disease, determining what crop to plant to best maximise the soil nutrient, pest control and management or knowing how much or how well their crops will perform, how the farmer solves these problems will determine the crop yield.

An accurate machine learning model uses several factors to help farmers improve their crop yield and make informed management and financial decisions, it also help policy makers to make import and export decisions (Khaki and Wang 2019).

1.2 Project description - briefly explain your project:

The proposed project will contain the development of five systems that contribute to improving crop yield. firstly, a construction of a machine learning system that predicts the best type of crops to be grown on soil based on the quantity of nitrogen, phosphorous, potassium and other factors present in the soil. Secondly a construction of machine learning system that predicts plant disease on farmland. The third system is a weather prediction system, it gives information about precipitation and other weather conditions in the area. The fourth system will be used for

---

pest control and management that avoid total reliance on pesticides, it uses image processing to detect pest on farmland, The fifth system is a weed management system, it is used in early detection of weeds on farmland.

[1] https://libguides.bournemouth.ac.uk/study-skills-referencing-plagiarism

The development of these systems would help farmers improves there crop yield.
The Proposed project will also contain a comprehensive review of how machine learning is being used in agriculture, a systematic literature review of climate-smart agriculture practises and a review of precision agriculture as well as their benefit and limitations.

1.3 Background - please provide brief background information, e.g., client, problem domain, referencing literature (minimum 4-5 sources):

Agriculture plays a critical role in our society; it is an important factor in the economic growth of a country (Bhanumathi et al. 2019).
The global population has grown from 1 billion in 1800's to 7.9 billion in 2020's, this growing population and an increase in scarcity of agricultural resources is driving demand for high quality and quantity of food.
Artificial intelligence, precision farming, and other advances in agricultural technologies have been helping farmers to fulfil this demand.
Machine learning is a branch of artificial intelligence that analyses data to discover pattern to make predictions, machine learning has been used in various fields including finance, business and Agriculture (van Klomenburg et al. 2020).
In agriculture machine learning has been used to boost the growth and yield of crops, it is used to predict crop yield, weather and rainfall, make crop and fertilizer recommendations etc. Crop yield prediction is based on different data collected and analysed to improve crop yield, it is influenced by many determinates considered as inputs including agricultural practices, biotic factors like pest and disease, and environmental factors like weather and soil properties, improving crop yield requires the understanding these determinates. (Liliane and Charles 2020; meneka and yuvaraj 2016).
Crop yield prediction models can help farmers generate more revenue and help improve quality of agriculture products(Dahikar and Rode 2014).

1.4 Research question(s):

The research question for the proposed project include:
What is the impact of climate change on agriculture?
What is the risk and adoption level of climate-smart agriculture?
What limitations do farmers face in precision agriculture?
What are the challenges in the field of crop yield prediction?
How can Artificial intelligence be used to improve crop yield?
What feature have been used in literature for crop yield predictions using machine learning?

1.5 Aims and objectives – should be <u>specific</u> and <u>measurable</u>:

The aim of the proposed project is to use publicly available dataset to develop a predictive model for improving crop yield.

The objectives of the proposed project include:

Objective 1: present an overview of machine learning applications in agriculture

Conduct a review on machine learning applications in agriculture, its advantage and limitations

Review the current research on improving crop yield

Objective 2: Conduct a review of agricultural practises

Conduct a review of climate change on agriculture

Conduct a review of climate-smart agriculture practises

Conduct a review of precision agriculture

Objective 3: Develop a predictive model for improving crop yield

Experiment with different dataset

Employ methods to test and evaluate the performance of predictive models for all systems

Define the best suited predictive model for all systems to improve crop yield

To evaluate the model's validity

Section 2: Artefact and Planning

2.1 What is the artefact that you intend to produce?

For this proposed project, the artefact is the development of five system that contribute to improving crop yield productivity. The five system are a crop recommendation system, a plant disease prediction system, a weather prediction system, a pest control and management system and a weed management system.

2.2 How do you intend on conducting the project (to produce the above artefact successfully)? *Please provide a GANTT chart at the end of this form, in Section 6.*

To be able to able to develop the artefact for this proposed project, licensed data will be sourced from various online data repositories such as Kaggle.

Various machine learning algorithms will be applied to each of the system to form a predictive model that analyses data to uncover patterns to improve crop yield prediction.

Section 3: Evaluation

3.1 How are you going to evaluate your project artefact?

The project report will feature an evaluation section that will outline how the artefact will be evaluated using various performance metrics such as Accuracy, precession rate, Recall, and Root mean Square Error.

Performance comparison between algorithms will also be carried out on each system **3.2 How does this project relate to your MSc Programme?**

As part of my programme MSC Data Science and artificial intelligence, this project is a complex, real-world problem that focuses on artificial intelligence area of my programme.

3.3 What are the risks in this project and how are you going to manage them?

It is foreseeable that in the conduction of this research, there are foreseeable privacy risk. In line with this, I, the researcher will only use publicly available, licenced data and comply with the licencing terms of the dataset provider.

Researcher bias will be avoided by ensuring all deductions and recommendations are evidencebased and necessary academic attribution will be done through intext citation and references.

Section 4: References

4.1 Please provide references for citations used above.

Bhanumathi, S., M. Vineeth, and N. Rohit. 2019 Crop yield prediction and efficient use of fertilizers. *International Conference on Communication and Signal Processing (ICCSP)*.

Dahikar, S. S., & Rode, S. V. (2014). Agricultural crop yield prediction using artificial neural network approach. *International journal of innovative research in electrical, electronics, instrumentation and control engineering*, *2*(1), 683-686.

Khaki, S. and Wang, L., 2019. Crop Yield Prediction Using Deep Neural Networks. *Frontiers in Plant Science*, 10.

Liliane, T. N., & Charles, M. S. 2020. Agronomy-Climate Change & Food Security. *Chapter, Factors Affecting Yield of Crops Factors Affecting Yield of Crops| IntechOpen*.

Menaka, K., and Yuvaraj, N. 2016. A survey on crop yield prediction models. *Indian Journal of Innovations and Developments*, *5*(12), 1-7.

van Klompenburg, T., Kassahun, A. and Catal, C., 2020. Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, 177, 105709.

Section 5: Academic Practice and Ethics

Please delete as appropriate.

5.1 Have you made yourself familiar with, and understand, the University guidance on referencing and plagiarism? **Yes**

5.2 Do you acknowledge that this project proposal is your own work and that it does Yes not contravene any academic offence as specified in the University's regulations?

5.3 Have you submitted online ethics checklist to your supervisor? Yes

5.4 Has the checklist been approved by your supervisor? No

Section 6: Proposed Plan (please attach a GANTT chart below)