Enhancing Crop yield prediction using Hybrid Neural Network models

CO8811- PROJECT REPORT

Submitted by

DIVAKAR R (211420118016)

VINOTHKUMAR S (211420118060)

in partial fulfillment for the award the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER AND COMMUNICATION ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

BONAFIDE CERTIFICATE

Certified that this project report **"Enhancing Crop yield prediction using Hybrid Neural Network models"** is the bonafide work of **DIVAKAR R (211420118016), VINOTHKUMAR S (211420118060)** who carried out the project work under my supervision.

SIGNATURE

Dr.B.ANNI PRINCY M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR,
COMPUTER AND COMMUNICATION ENGINEERING,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI, POONAMALLEE,
CHENNAI- 600123.

SIGNATURE

V.MUTHU M.Tech.,(Ph.D.),

SUPERVISOR

ASSOCIATE PROFESSOR,
ARTIFICIAL INTELLIENGE AND MACHINE LEARNING,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI, POONAMALLEE,
CHENNAI- 600123.

Certified that the above candidate(s) was/ were examined in the End Semester Major Project Viva-Voce Examination held on.....

INTERNALEXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors Tmt. **C.VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D**., and **Dr.SARANYASREE SAKTHIKUMAR B.E., M.B.A., Ph.D**., for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CCE Department, **Dr. B. ANNI PRINCY, M.E.,Ph.D**., Professor, for the support extended throughout the project.

We would like to thank my supervisor **V.MUTHU.,M.Tech.,(Ph.D**.)., Assistant Professor and all the faculty members of the Department of Artificial intelligence and machine learning for their advice and suggestions for the successful completion of the project.

VINOTHKUMAR S

DIVAKAR R

ABSTRACT

Traditional methods often rely on historical data and statistical models, which may lack accuracy and robustness in capturing the complex interactions between various environmental factors and crop growth dynamics. Using Random Forest techniques have shown promising results in improving crop yield prediction by leveraging large-scale datasets. This paper presents an overview of the application of Random Forest techniques for crop yield prediction. We discuss various factors affecting crop yield, including weather conditions, soil properties, crop types, and agricultural practices, and explore how Random Forest models can effectively integrate these factors to make accurate predictions. Furthermore, we compare other types of algorithms commonly used in crop yield prediction, such as Linear regression, Gradient Boost, XG Boost, KNN, Decision Tree, Bagging Regressor. are classification. Additionally, we highlight the challenges and opportunities in implementing Random forest approaches for crop yield prediction, including Accuracy, MSE, R2 Score and generalization across different regions and crop types. We discuss potential solutions to address these challenges and suggest future research directions to further improve the accuracy and applicability of Random Forest Based crop yield prediction models.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN - Convolution Neural Network

DNN - Deep Neural Network

HTML - Hypertext Markup Language

CSS - Cascading Style Sheet

SVM - Support Vector Machine

RFA - Random Forest Algorithm

CHAPTER 1

INTRODUCTION

1.1 Introduction

The agriculture industry has used modern science to meet the food demands of 7 billion people. However, there are numerous threats that people working in the agriculture industry face that threaten the food security of the human society. Some of the threats as we know include, climate change, livestock grazing, plant diseases, etc. Among the many threats, the effect of plant destroyed by rainfall is truly momentous as it not only causes huge wastage of plants for human consumption but it also immensely affects the health of the human society and the lives of the farmers whose main source of income is from their production of healthy crops. During the process of plant harvesting, human experts go through a tedious process of checking and removing mature plants, making sure they aren't affected by any disease and are suitable for human consumption.

Identification of plant diseases has been made much easier, less time consuming and cheaper in comparison to the traditional visual identification of plant diseases. A lot of research has been carried out in this domain in recent years, because of which the industry is slowly moving towards replacing the traditional identification of plant diseases with machine learning models .The aim of this thesis is to implement two different machine learning models, namely, Random Forest Algorithm dataset and also evaluate the aforementioned models based on the following evaluation metrics: Accuracy, Precision, Recall and R2-Score. Agriculture, being inherently susceptible to various environmental, biological, and socio-economic factors, has historically grappled with uncertainties in yield estimation. Traditional methods relying on historical data and intuition often fall

short in accurately forecasting crop yields, leading to inefficiencies in resource management, market instability, and food insecurity. However, the advent of cutting-edge technologies such as machine learning, remote sensing, and big data analytics has catalyzed a transformation in this landscape, offering novel avenues to predict crop yields with unprecedented precision. At its core, crop yield prediction serves as a powerful tool to optimize agricultural operations across the entire value chain. For farmers, it offers invaluable insights into optimal planting schedules, resource allocation (such as water, fertilizers, and pesticides), and risk management strategies. By aligning cultivation practices with predicted yields, farmers can enhance crop resilience, minimize input costs, and maximize profitability. Moreover, for policymakers and market analysts, accurate yield forecasts facilitate informed decision-making regarding trade policies, food security initiatives, and infrastructure development, thereby fostering sustainable agricultural growth and economic stability. models and machine learning algorithms to remote sensing techniques. Statistical models leverage historical data on weather patterns, soil characteristics, crop varieties, and agronomic practices to generate yield projections. Meanwhile, machine learning algorithms harness the power of vast datasets to discern intricate patterns and correlations, thereby offering more nuanced predictions. Remote sensing technologies, including satellite imagery and drones, provide real-time insights into crop health, growth stages, and environmental conditions, augmenting the accuracy and timeliness of yield forecasts. In essence, crop yield prediction heralds a new era of precision agriculture, wherein data-driven insights drive sustainable and efficient farming practices. As we navigate the complexities of a rapidly changing climate and burgeoning global population, the ability to anticipate crop yields assumes paramount importance in ensuring food security, economic prosperity, and environmental stewardship. By embracing innovation and collaboration, we can harness the transformative potential of crop yield prediction to cultivate a brighter future for generations to come.

1.2 Existing System

Traditional statistical models, such as linear regression and time series analysis, have been widely used for crop yield prediction. These models rely on historical data related to crop yields, weather patterns, soil characteristics, and agronomic practices to make forecasts. While simple and interpretable, they may struggle to capture complex non-linear relationships and are often limited by the availability of high-quality data. Remote Sensing: Remote sensing technologies, including satellite imagery and aerial drones, provide valuable information about crop health, vegetation indices, and environmental conditions. By analyzing these data sources using techniques like spectral analysis and machine learning algorithms, researchers can estimate crop yields with a high degree of accuracy. Remote sensing offers the advantage of scalability and real-time monitoring, allowing for timely interventions and adaptive management strategies. Machine Learning: Machine learning algorithms, such as random forests, support vector machines, and neural networks, have gained prominence in crop yield prediction due to their ability to handle large and complex datasets. By training models on historical yield data along with relevant features like weather variables, soil properties, and crop management practices, machine learning techniques can identify intricate patterns and make precise predictions. These models can be continuously refined and updated as new data becomes available, improving their predictive performance over time. Integrated Decision Support Systems: Integrated decision support systems combine multiple data sources, models, and algorithms to provide comprehensive insights into crop yield prediction. These systems leverage a holistic approach, incorporating factors such as weather forecasts, soil moisture levels, pest and disease outbreaks, market trends, and socioeconomic indicators to generate actionable recommendations for farmers and policymakers. By integrating diverse sources of information, decision support systems enable stakeholders to make informed decisions and optimize resource

allocation for improved agricultural outcomes. While each of these methods has its strengths and limitations, the ongoing advancements in technology and data science hold the promise of further enhancing the accuracy and reliability of crop yield prediction systems. By harnessing the synergies between different approaches and fostering interdisciplinary collaboration, researchers and practitioners can continue to drive innovation in agricultural forecasting and contribute to the sustainable intensification of food production.

1.3 Proposed System

In this research Gather historical data on crop yields, weather patterns, soil quality, pest and disease outbreaks, crop rotation practices, and other relevant factors from various sources like government databases, agricultural research institutions, and satellite imagery. Data Preprocessing: Cleanse the collected data, handle missing values, normalize or scale features, and perform feature engineering to extract meaningful insights. Identify the most relevant features that significantly impact crop yields using techniques like correlation analysis, feature importance, or domain knowledge. Model Selection: Choose appropriate machine learning algorithms such as regression, decision trees, random forests, or neural networks based on the nature of the data and the problem at hand.Model Training: Train the selected models on the historical data using techniques like cross-validation to optimize hyperparameters and prevent overfitting. Evaluate the performance of trained models using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) on a separate validation dataset. Evaluate the performance of trained models using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) on a separate validation dataset. Integration with External Systems: Integrate the prediction system with other agricultural management systems or decision support tools to provide actionable insights to farmers or agricultural stakeholders. Monitoring and Maintenance: Continuously monitor the

performance of the deployed model, retrain it periodically with new data to keep it up-to-date, and make necessary adjustments based on feedback from users or changes in the environment. One of the important sectors of Indian Economy is Agriculture. Employment to almost 50% of the countries workforce is provided by Indian agriculture sector.India is known to be the world's largest producer of pulses, rice, wheat, spices and spice products. Farmer's economic growth depends on the quality of the products that they produce, which relies on the plant's growth and the yield they get. Plants are highly prone to diseases that affect the growth of the plant which in turn affects the ecology of the farmer. In order to detect a plant disease at very initial stage, use of automatic disease detection technique is advantageous. Manual detection of plant disease using leaf images is a tedious job. Hence, it is required to develop computational methods which will make the process of disease detection and classification using leaf images automatic.

1.4 Problem Statement

Agriculture plays a pivotal role in global food security and economic stability. However, predicting crop yields accurately remains a challenging task due to the complex interplay of various factors such as weather conditions, soil quality, crop type, and management practices. Accurate crop yield prediction is essential for farmers, policymakers, and stakeholders to make informed decisions regarding crop planning, resource allocation, and market strategies.The objective of this project is to develop a robust machine learning model that can accurately predict crop yields based on historical data and relevant environmental variables. The model should be capable of providing timely and reliable forecasts for different crops and regions, helping stakeholders mitigate risks and optimize agricultural practices.

1.5 Objectives

The motivation for including explain ability lies in the fact that most of the ma- chine learning models widely used in this domain are black-box models, which leads to the users using these models for prediction in not trusting and under- standing how their models. A lot of the research that has been carried out on plant disease detection present a comparative study using different machine learning models but fail to explain the predictions made by their models. In this research we not only provide a comparative study between the workings of a simple and complex mode.

1.6 Methodology

1.6.1 Data Collection and Preprocessing

Gather historical data on crop yields, weather patterns, soil characteristics, and agronomic practices from reliable sources. Cleanse and preprocess the data to handle missing values, outliers, and inconsistencies.

Model Selection

Choose appropriate machine learning algorithms based on the characteristics of the data and the problem at hand. Commonly used algorithms include linear regression, decision trees, random forests, support vector machines, and neural networks.

Training-Validation Split

Split the dataset into training and validation sets to train the model on one subset and evaluate its performance on another. This helps assess the model's generalization ability and identify potential overfitting.

Model Training

Train the selected model(s) on the training data using techniques like cross-validation, grid search, or random search to find optimal hyperparameters. This involves fitting the model to learn the relationship between input features and crop yields.

Feature Engineering and Selection

Explore various feature engineering techniques to extract relevant information from raw data. Employ statistical analysis and domain knowledge to identify the most influential features for crop yield prediction.

Model Development

Experiment with a range of machine learning algorithms such as regression models, decision trees, random forests, and neural networks. Tune hyperparameters and evaluate model performance using appropriate metrics such as mean absolute error (MAE) or root mean square error (RMSE).

Cross-Validation and Evaluation

Validate the model using cross-validation techniques to assess its generalization ability. Compare the performance of different models and select the best-performing one based on predefined criteria.

Model Selection

Choose appropriate machine learning algorithms such as regression, decision trees, random forests, or neural networks based on the nature of the data and the problem at hand.

Model Tuning

Fine-tune the model by adjusting hyperparameters or trying different algorithms to improve performance. This iterative process may involve experimentation and validation to find the best-performing configuration.

Prediction

Once the model is trained and validated, use it to make predictions on new or unseen data. This involves applying the trained model to input features to generate predictions of crop yields.

Model Interpretation

Interpret the results of the model to understand the factors contributing to crop yield predictions. This helps provide insights into the relationships between input variables and crop yields, guiding decision-making processes for farmers or agricultural stakeholders.

Deployment

Deploy the trained model into production, either as a standalone application, integrated into existing agricultural systems, or accessible via APIs. Ensure scalability, reliability, and real-time or batch processing capabilities as needed.

1.6.2 Neural Network

Neural networks are increasingly being used in climate prediction due to their ability to handle complex, non-linear relationships in data. Here's a general methodology for using neural networks in climate prediction:

Data Collection

Collect historical climate data from various sources, including temperature, precipitation, humidity, wind speed, and other relevant variables. This data is typically collected over a long period to capture seasonal and long-term trends.

Data Preprocessing

Clean the data to remove noise and outliers, and preprocess it for use in the neural network. This may involve normalizing the data, handling missing values, and encoding categorical variables.

Feature Selection/Extraction

Select relevant features (input variables) that are likely to have a significant impact on climate prediction. This can be done manually based on domain knowledge or using automated feature selection techniques.

Model Selection

Choose a suitable neural network architecture for climate prediction. This could be a simple feedforward neural network, a convolutional neural network (CNN) for spatial data, or a recurrent neural network (RNN) for temporal data.

Model Training

Split the data into training, validation, and test sets. Train the neural network using the training set and validate it using the validation set. Adjust hyperparameters such as learning rate, batch size, and number of epochs to optimize the model performance.

Model Evaluation

Evaluate the trained model using the test set to assess its performance. Common evaluation metrics for climate prediction include mean squared error (MSE), mean absolute error (MAE), and correlation coefficient.

Prediction

Once the model is trained and evaluated, use it to make predictions on new data. Monitor the model's performance over time and retrain it periodically to maintain its accuracy.

Uncertainty Estimation

Climate prediction is inherently uncertain due to the complex nature of climate systems. Consider using techniques such as dropout or ensemble to estimate uncertainty in the predictions.

Post-processing

Post-process the model outputs if necessary to derive useful information for decision-making. This could involve aggregating predictions over regions or time periods, or translating them into actionable insights.

Communication

Communicate the model results and uncertainties effectively to stakeholders, policymakers, and the public to support informed decision-making. This methodology provides a general framework for using neural networks in climate prediction, but specific implementations may vary based on the nature of the data and the goals of the prediction task.

1.6.3 Support Vector Machine

Identifying the most relevant features is crucial for the performance of the SVM model. Feature selection techniques or domain knowledge can be used to determine which features have the most impact on crop yield.After preprocessing the data, it's divided into training and testing sets. The SVM model is then trained on the training set. During training, the SVM algorithm finds the optimal

hyperplane that best separates the data points into different classes, in this case, different levels of crop yield.

Once the model is trained, it needs to be evaluated using the testing set to assess its performance. Common evaluation metrics for regression tasks, such as mean squared error (MSE), root mean squared error (RMSE), or coefficient of determination (R-squared), can be used to measure the accuracy of the predicted crop yields compared to the actual yields.SVM has parameters like kernel type, regularization parameter (C), and kernel coefficient (gamma) that need to be tuned for optimal performance. Techniques like grid search or random search can be used to find the best combination of hyperparameters.Once the model is trained and evaluated, it can be used to predict crop yields for new data. The model takes input features such as weather conditions, soil characteristics, etc., and predicts the corresponding crop yield. Crop yield prediction is a complex task influenced by various factors. It may require iterative improvement by incorporating additional data sources, refining feature selection, or experimenting with different machine learning algorithms.SVMs have been successfully used in agriculture for various tasks including crop yield prediction, disease detection, and classification of crop types

1.7 Language Used

Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. Python has a syntax that emphasizes readability and reduces the cost of program maintenance. This makes it an excellent choice for beginners and experienced programmers alike. Python can be used for a wide range of applications, including web development, data analysis, artificial intelligence, scientific computing, automation, and more. Its extensive standard library and third-party packages make it suitable for various tasks.

Python is an interpreted language, meaning that code is executed line by line. This makes it easy to debug and test code interactively. Python uses dynamic typing, allowing variables to be assigned without specifying their type explicitly. This enhances flexibility but may require careful attention to variable types in larger projects.

Strong Community and Ecosystem: Python has a large and active community of developers who contribute to its ecosystem by creating libraries, frameworks, and tools. Popular libraries include NumPy, Pandas, Matplotlib, TensorFlow, Django, Flask, and many others.Python is open-source, meaning that its source code is freely available and can be modified and distributed by anyone. This fosters innovation and collaboration within the Python community.Python is available on multiple platforms, including Windows, macOS, and various Unix-based operating systems, making it accessible to a wide range of users.

HTML

HTML (Hypertext Markup Language) is the standard markup language used to create web pages. HTML provides the structure for web pages by using elements and tags. Elements are enclosed within angle brackets , and most come in pairs: an opening tag and a closing tag. The content is placed between these tags. For example, denotes a paragraph tag, and denotes the closing tag for the paragraph.

CSS

CSS, or Cascading Style Sheets, is a language used for describing the presentation of a document written in HTML or XML, including aspects like layout, colors, fonts, and animations. It allows web developers to control the appearance of web pages, ensuring consistency across different devices and browsers. CSS works by selecting HTML elements and applying styling rules to them, either directly within HTML files or externally in separate CSS files. It

follows a cascading hierarchy where rules can be inherited, overridden, or combined to achieve the desired styling effects.

1.7.1 System Configuration

H/W System Configuration

RAM	-	4 GB (min)
SOFTWARE ID	-	python idle
Mouse	-	Two or Three Button Mouse

S/W SYSTEM CONFIGURATION

Operating System	-	Windows 8/10
Front End	-	CSS
Language	-	Python(3.10) Version.
Server	-	Flask.

1.8 Organization

International Food Policy Research Institute (IFPRI) conducts research on agricultural productivity, food security, and economic development. They develop models and tools for crop yield prediction to inform policy decisions and promote sustainable agriculture.National Aeronautics and Space Administration (NASA) utilizes satellite imagery and remote sensing data to monitor agricultural conditions worldwide. They collaborate with research organizations and government agencies to develop crop yield prediction models based on these data.United States Department of Agriculture (USDA) conducts extensive

research on crop production, yield forecasting, and agricultural economics. They publish reports and provide data-driven insights to farmers, policymakers, and stakeholders to support decision-making.European Commission's JRC develops models and tools for crop yield prediction and agricultural monitoring using satellite data and machine learning techniques.

They provide valuable information and services to support European Union policies related to agriculture and food security. National Agricultural Research Systems (NARS) Each country typically has its own NARS, which conducts research, develops technologies, and provides extension services to farmers aimed at improving crop productivity and resilience. Private sector seed companies play a significant role in developing and disseminating high-yielding crop varieties, often through hybridization and genetic modification.Non-Governmental Organizations (NGOs) such as the Bill & Melinda Gates Foundation, Oxfam, and CARE International are involved in various agricultural development projects aimed at increasing crop yields, particularly in regions facing food insecurity.Farmers' Cooperatives and Associations: These grassroots organizations often collaborate with government agencies, NGOs, and research institutions to access improved seeds, technologies, and agronomic practices that can enhance crop yields.Academic and Research Institutions Universities and research centers conduct fundamental and applied research aimed at understanding crop biology, genetics, and agronomy to develop innovative solutions for improving crop yields sustainably. Ministries or departments of agriculture play a vital role in formulating policies, providing extension services, and implementing programs to support farmers in improving crop productivity and yield.

CHAPTER 2

LITERATURE SURVEY

Title 1: Estimation Of Leaf Angle Distribution Based On Statistical Properties Of Leaf Shading Distribution

Author: kuniaki uto , mauro dalla mura , yuka sasaki and koichi shinoda

Year: 2020

Description:

Leaf angle distribution is an important phenotype parameter that is related to photosynthesis. Thanks to the recent advent of drones and high-resolution imaging devices, leaf-scale aerial images with high spectral and spatial resolution are available. This work is the first attempt to utilize a single leafscale image to differentiate plants with different leaf angle distribution. First, assuming that a rice leaf surface resembles a section of a hemiellipsoid surface, a collection of rice leaf surfaces is approximated by a hemiellipsoid surface. investigating the statistical properties, i.e., skewness, kurtosis and the most probable intensity, of the frequencies of the simulated shading intensity that welldifferentiate hemiellipsoids with different structural parameters, we identified an appropriate time slot, i.e., 11:00-12:30, for image acquisitions. Then, time-series leaf-scale images and depth maps of rice plants with/without silicate fertilizer under sunlight were collected. Based on the depth maps, it was confirmed that silicate fertilizer dosed leaves are more upright than leaves from non treated plants. It was demonstrated that 89% and 100% of kurtosis and the most probable intensity of the leaf-scale images during the appropriate time slot showed consistent relations with the simulations, which indicates that the proposed method is useful to

distinguish different leaf angle distributions based on the frequency of shading intensity of rice leaf images.

Title 2: Plant Species Identification From Occluded Leaf Images

Author: ayan chaudhury, and john l. barron

Year: 2020

Description:

We present an approach to identify the plant species from the contour information from occluded leaf image using a database of full plant leaves. Although contour based 2D shape matching has been studied extensively in the last couple of decades, matching occluded leaves with full leaf databases is an open and little worked on problem. Classifying occluded plant leaves is even more challenging than full leaf matching because of large variations and complexity of leaf structures. Matching an occluded contour with all the full contours in a database is an NP-hard problem, so our algorithm is necessarily suboptimal. First, we represent the 2D contour points as a β-Spline curve. Then we extract interest points on these curves via the Discrete Contour Evolution (DCE) algorithm. We use subgraph matching using the DCE points as graph nodes, which produces a number of open curves for each closed leaf contour. Next, we compute the similarity transformation parameters (translation, rotation and uniform scaling) for each open curve. We then "overlay" each open curve with the inverse similarity transformed occluded curve and use the Frechet ´ distance metric to measure the quality of the match, retaining the best η matched curves. Since the Frechet metric is cheap to compute ´ but not perfectly correlated with the quality of the match, we formulate an energy functional that is well correlated with the quality of the match, but is considerably more expensive to compute. The functional uses local and global curvature, Shape Context descriptors and String

Cut features. We minimize this energy functional using a convex-concave relaxation framework.

Title 3: Tomato Septoria Leaf spot necrotic and chlorotic regions computational assessment using artificial bee colony-optimized leaf disease index

Author: ronnie concepcion, sandy lauguico, elmer dadios, argel bandala, edwin sybingco, jonnel alejandrino

Year: 2020

Description:

Visual inspection of plant health status and disease severity may yield subjective assessments due to errorprone sphere of colors and textures as affected by angular photosynthetic light source and the complexity of chlorosis. Quantification of damages on leaves due to destructive diseases is paramount for plant and pathogen interactions. To address this challenge, the proposed solution is the integration of computer vision and computational intelligence for tomato Septoria leaf spot necrotic and chlorotic region computational assessment. Dataset contains healthy and diseased tomato leaves that were captured individually. Non-vegetation pixels removal was done using CIELab color space. RGB color components and five Haralick texture features were extracted from the segmented leaf. Hybrid neighborhood component analysis and ReliefF algorithm were employed to select the important predictors resulting to RGB-entropy vector. A new tomato leaf disease index (tomLDI) optimized using artificial bee colony (ABC) was developed by normalizing visible red reflectance, and introducing red-green and red-blue reflectance ratios to enhance Septoria leaf spots pixels and reducing sensitivity to healthy green pixels. KNN bested classification tree, linear discriminant analysis and Naïve Bayes in detecting Septoria leaf disease with accuracy of 97.46%. Deep transfer image regression was tested using raw infected leaf images and the tomLDI transformed colored channels through MobileNetV2, ResNet101 and InceptionV3. Using tomLDI channel, MobileNetV2 and

ResNet101 bested other networks in estimating leaf diseased region percentage and number of Septoria spots with R2 values of 0.9930 and 0.9484 respectively. tomLDI channel proved to be more accurate than using raw images for regression.

Title 4: Tomato Leaf Disease Identification By Restructured Deep Residual Dense Network

Author: changjian zhou, sihan zhou, jinge xing, and jia song

Year: 2021

Description:

As COVID-19 spread worldwide, many major grain-producing countries have adopted measures to restrict their grain exports, food security has aroused great concern from various parties. How to improve grain production has become one of the most important issues facing all countries. However, crop diseases are a difficult problem for many farmers so it is important to master the severity of crop diseases timely and accurately to help staff take further intervention measures to minimize plants being further infected. In this paper, a restructured residual dense network was proposed for tomato leaf disease identification; this hybrid deep learning model combines the advantages of deep residual networks and dense networks, which can reduce the number of training process parameters to improve calculation accuracy as well as enhance the flow of information and gradients. The original RDN model was first used in image super resolution, so we need to restructure the network architecture for classification tasks through adjusted input image features and hyper parameters. Experimental results show that this model can achieve a top-1 average identification accuracy of 95% on the Tomato test dataset in AI Challenger 2018 datasets, which verifies its satisfactory performance.

Title 5: Classification Of Watermelon Leaf Diseases Using Neural Network Analysis

Author: suhaili beeran kutty,noor ezan abdullah, dr. hadzli hashim

Year: 2020

Description:

This paper mainly discussed the process to classify Anthracnose and Downey Mildew, watermelon leaf diseases using neural network analysis. A few of infected leaf samples were collected and they were captured using a digital camera with specific calibration procedure under controlled environment. The classification on the watermelon's leaf diseases is based on color feature extraction from RGB color model where the RGB pixel color indices have been extracted from the identified Regions of Interest (ROI). The proposed automated classification model involved the process of diseases classification using Statistical Package for the Social Sciences (SPSS) and Neural Network Pattern Recognition Toolbox in MATLAB. Determinations in this work have shown that the type of leaf diseases achieved 75.9% of accuracy based on its RGB mean color component.

CHAPTER 3

SYSTEM DEVELOPMENT

System development typically refers to the process of creating or improving systems, which are sets of interconnected components working together to achieve a common goal. This process involves various stages, such as planning, designing, implementing, testing, and maintaining the system. It often includes activities like gathering requirements from stakeholders, designing the system architecture, developing the software, and deploying the system. System development can be applied to various types of systems, including software systems, information systems, and hardware systems, among others.

3.1 Function Block Diagram of the proposed system

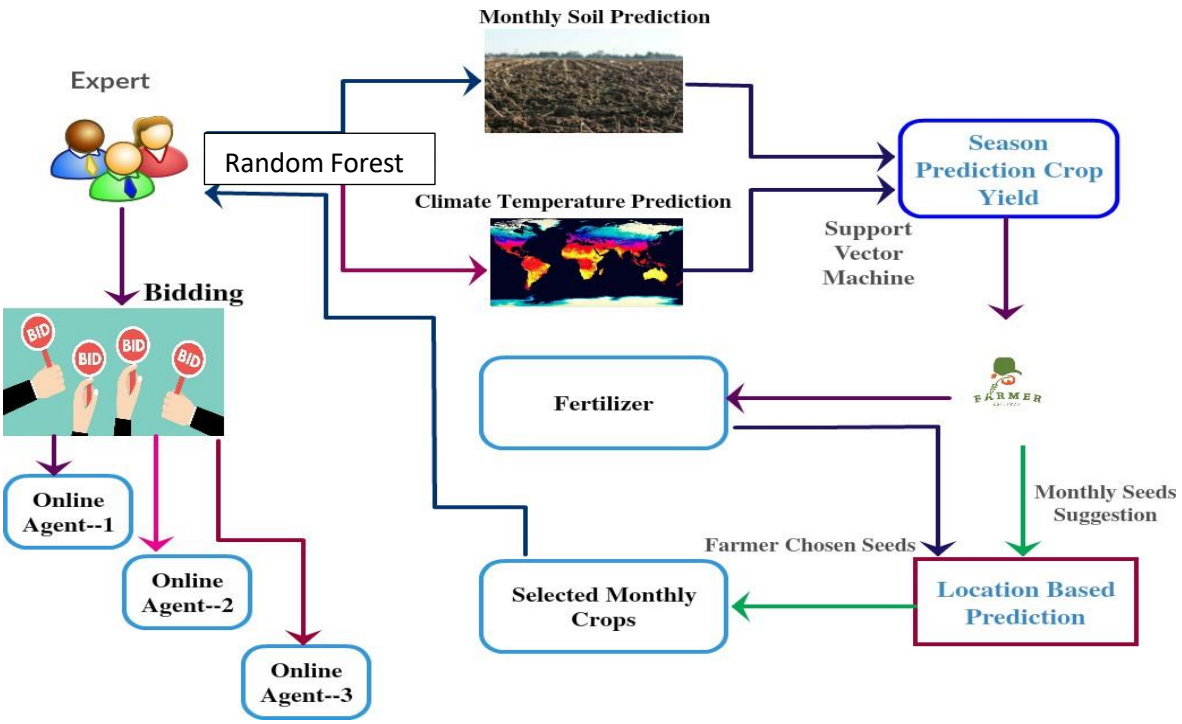


Figure:3.1 Functional Block Diagram

Support Vector Machine

Identifying the most relevant features is crucial for the performance of the SVM model. Feature selection techniques or domain knowledge can be used to determine which features have the most impact on crop yield.After preprocessing the data, it's divided into training and testing sets. The SVM model is then trained on the training set. During training, the SVM algorithm finds the optimal hyperplane that best separates the data points into different classes, in this case, different levels of crop yield.

Once the model is trained, it needs to be evaluated using the testing set to assess its performance. Common evaluation metrics for regression tasks, such as mean squared error (MSE), root mean squared error (RMSE), or coefficient of determination (R-squared), can be used to measure the accuracy of the predicted crop yields compared to the actual yields.SVM has parameters like kernel type, regularization parameter (C), and kernel coefficient (gamma) that need to be tuned for optimal performance. Techniques like grid search or random search can be used to find the best combination of hyperparameters.Once the model is trained and evaluated, it can be used to predict crop yields for new data. The model takes input features such as weather conditions, soil characteristics, etc., and predicts the corresponding crop yield. Crop yield prediction is a complex task influenced by various factors. It may require iterative improvement by incorporating additional data sources, refining feature selection, or experimenting with different machine learning algorithms.SVMs have been successfully used in agriculture for various tasks including crop yield prediction, disease detection, and classification of crop types

Climate Temperature Prediction

A comprehensive crops dataset encompasses a plethora of information crucial for agricultural planning and analysis. It includes details on various crops

such as wheat, rice, corn, soybeans, potatoes, tomatoes, cotton, barley, sugarcane, and sunflowers, among others. Each crop entry in the dataset contains essential attributes like growth period, Random Forest Algorithm used for soil type preference, temperature range tolerance, water requirements, and yield potential. For instance, wheat typically thrives in loamy soil with a growth period spanning 90 to 120 days, while requiring moderate water and flourishing in temperatures ranging from 10°C to 24°C. On the other hand, crops like sugarcane have a significantly longer growth period of 12 to 24 months, preferring well-drained soil and thriving in temperatures between 20°C to 30°C with high water needs. This dataset serves as a valuable resource for farmers, researchers, and policymakers, facilitating informed decision-making regarding crop selection, resource allocation, and agricultural planning strategies to optimize yield and ensure food security.

Monthly Soil Prediction

Gather data from various sources such as agricultural research institutions, government agencies, satellite imagery, weather stations, and field surveys. This data may include historical crop yield records, weather data, soil characteristics, satellite imagery, crop management practices, and socio-economic factors.Clean the collected data to address missing values, outliers, inconsistencies, and errors. This may involve techniques such as imputation for missing values, outlier detection and removal, and validation against known standards or domain knowledge.Feature Identify relevant features using Random Forest Algorithm (variables) that are likely to influence crop yield prediction. This may involve domain knowledge, statistical analysis, and feature engineering techniques such as principal component analysis (PCA), feature scaling, and transformation to derive new features. Data Integration: Integrate data from different sources and formats into a unified dataset. Ensure consistency in data formats, units, and spatial/temporal resolutions. Normalize or standardize the data to bring all

features to a similar scale. This step is crucial for algorithms sensitive to the scale of features, such as neural networks and support vector machines. Split the dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperparameters and evaluate model performance during training, and the test set is used to evaluate the final model's performance. If dealing with time-series data handle temporal dependencies appropriately. Consider techniques such as lagging variables, rolling window statistics, and time-series decomposition.If the dataset contains categorical variables encode them into numerical format using techniques like one-hot encoding or label encoding.

Season Prediction Crop Yield

Feature extraction is a critical step in preparing data for predictive modeling, including crop yield prediction. Previous crop yields from the same field or region.Average yield over multiple years to capture long-term trends.Yield variability and standard deviation to account for fluctuations.Average temperature during different growth stages.Total precipitation or rainfall distribution throughout the growing season.Growing degree days (GDD) calculated based on temperature thresholds for crop growth.Climate anomalies or extreme events (e.g., droughts, heatwaves) and their impacts on yield.Soil pH, organic matter content, and nutrient levels.Soil texture and water-holding capacity.Depth to bedrock or water table, which affects root penetration and water availability. Planting date and duration of the growing season.Fertilizer application rates and timing.Irrigation frequency, volume, and method Crop rotation history and previous crops grown in the same field.Crop type, variety, and genetic traits (e.g., drought tolerance, disease resistance).Phenological stages such as emergence, flowering, and maturity.Pest and disease pressure and management practices Normalized Difference Vegetation Index (NDVI) to assess

crop health and vigor.Leaf Area Index (LAI) or canopy cover to quantify biomass accumulation.Surface temperature or thermal imaging to monitor crop stress and water status.Geographic coordinates (latitude, longitude) of field locations.Elevation and slope of the terrain, which influence water drainage and erosion.Land use/land cover classification to identify crop types and landscape heterogeneity.Crop prices and market demand for forecasting economic returns.Input costs (e.g., seeds, fertilizers, fuel) and labor requirements.Government policies, subsidies, and trade agreements affecting agricultural markets.

Fertilizer

Fertilizer application plays a crucial role in predicting crop yield. By starting with soil testing to assess nutrient levels and pH, farmers can tailor their fertilizer choices and application rates to match the specific needs of their crops. Understanding the nutrient requirements of the crop at various growth stages is essential for determining the timing and quantity of fertilizer application. Additionally, considering environmental factors such as rainfall, temperature, and soil moisture levels helps optimize nutrient availability and uptake. Regular monitoring of crop health and growth allows for adjustments in fertilizer application as needed throughout the growing season. Integrating data-driven approaches and seeking guidance from agricultural experts further enhances fertilizer management practices, ultimately contributing to more accurate crop yield predictions and improved agricultural productivity.

Bidding

Bidding for crop yield prediction involves a multi-faceted approach that integrates various data sources and analytical methods. It begins with acquiring relevant data, including historical yield records, weather patterns, soil

characteristics, and agronomic practices. This data serves as the foundation for developing predictive models using statistical techniques, machine learning algorithms, or a combination of both. Bidders employ sophisticated modeling frameworks that account for the complex interactions between different variables to generate accurate yield forecasts. Additionally, incorporating remote sensing data, satellite imagery, and other geospatial information enables bidders to capture spatial variability and assess crop conditions at a finer scale. As the bidding process unfolds, participants may leverage their expertise in agronomy, data science, and market dynamics to refine their models and strategies.

Experts

Crop yield prediction requires expertise from various fields, including agronomy, data science, and agricultural engineering. Agronomists, with their understanding of crop physiology and environmental factors, play a crucial role in determining the optimal conditions for crop growth and predicting yield outcomes. Data scientists utilize statistical models and machine learning algorithms to analyze historical yield data, satellite imagery, weather patterns, and soil information to develop predictive models. Agricultural engineers contribute by integrating sensor technologies, precision agriculture techniques, and remote sensing tools to gather real-time data on crop health, soil conditions, and environmental variables.

Overall Future Crop Yield Prediction

Gather historical data on crop yields, weather patterns, soil characteristics, and other relevant variables for the target region.Acquire real-time or near-real-time data on current environmental conditions, such as temperature, precipitation, humidity, and solar radiation.Extract relevant features from the collected data, including past yield trends, seasonal patterns, soil moisture levels, temperature

variations, and other factors known to influence crop growth.Preprocess the data to handle missing values, outliers, and inconsistencies.Choose appropriate predictive modeling techniques suited for crop yield prediction, such as regression analysis, time series forecasting, machine learning algorithms hybrid models combining multiple approaches.Consider the scalability, interpretability, and computational requirements of the chosen models.Split the dataset into training and validation sets to assess model performance.Train the selected models on the historical data, using techniques like cross-validation to optimize hyperparameters and prevent overfitting.Evaluate the models' performance using metrics such as mean absolute error, root mean square error, or coefficient of determination (R-squared).Use the trained models to forecast future crop yields based on current environmental conditions and projected trends.Incorporate uncertainty estimates to provide confidence intervals or probabilistic forecasts, acknowledging the inherent variability and unpredictability of agricultural systems.Integrate advanced techniques like ensemble modeling or Bayesian inference to improve prediction accuracy and robustness.Validate the model predictions against actual yield data from subsequent growing seasons.Assess the model's reliability, recalibrating or fine-tuning as necessary to adapt to evolving environmental conditions or changing agricultural practices.Continuously update the dataset with new observations and iteratively improve the prediction models over time.Deploy the validated models as decision support tools for farmers, agricultural advisors, and policymakers.Provide user-friendly interfaces or visualizations to communicate forecasted yield trends, identify risk factors, and recommend adaptive management strategies.Foster collaboration between stakeholders to leverage predictive insights for sustainable agricultural planning, resource allocation, and risk mitigation efforts.

3.2 Modules

- Data Exploration and Analysis
- Data Preprocessing
- Linear Regression.
- Random Forest

3.3 Module Description

3.3.1 Data Exploration and Analysis

Begin by obtaining a comprehensive understanding of the dataset, including its size, structure, and variables. Identify the features available and their respective data types Understand the temporal and spatial coverage of the data, as crop yields may vary based on factors such as seasonality and geographical location.Compute descriptive statistics for numerical features, such as mean, median, standard deviation, minimum, and maximum values. This provides insights into the central tendency, spread, and distribution of the data.For categorical features, calculate frequency counts or proportions to understand the distribution of different categories.Generate visualizations to explore the relationships between variables and identify potential patterns or trends. Common plots include histograms, box plots, scatter plots, line plots, and heatmaps.Visualize the distribution of crop yields over time and across different regions. Time series plots can reveal seasonal patterns and long-term trends, while spatial maps can highlight regional variations.Explore the correlation between crop yield and various factors such as weather conditions, soil properties, agricultural practices, and socioeconomic indicators. Use scatter plots or correlation matrices to visualize pairwise relationship.Conduct temporal analysis to identify seasonal trends and patterns in crop yield data. Aggregate the data by time intervals and visualize the trends over time.Examine the influence of weather

variables on crop yield by analyzing their temporal patterns and correlations.Perform spatial analysis to assess the spatial variability of crop yields and its relationship with environmental factors. Use geographical information systems (GIS) tools to visualize crop yield distributions on maps and overlay them with spatial layers such as soil types, land cover, and elevation.Identify spatial clusters or hotspots of high or low crop yields and explore the underlying factors contributing to spatial patterns.

3.3.2 Data Preprocessing

Gain a clear understanding of the sources from which you'll be collecting data. This could include historical yield data, weather data, soil data, satellite imagery, and other relevant sources. Collect the data from various sources, ensuring it covers a significant timeframe and geographical area Preprocess the data to handle missing values, outliers, and inconsistencies. This may involve techniques such as imputation, outlier detection, and data normalization.Start by exploring basic statistics of your data, such as mean, median, standard deviation, and range. This can give you initial insights into the distribution and variability of your features.Visualize the data using plots such as histograms, box plots, scatter plots, and time series plots. This can help identify patterns, trends, and relationships between variables.Conduct correlation analysis to understand the relationships between different features. For example, you might want to see how weather variables correlate with crop yield.Explore the seasonality and trends in your data, especially for time series data. This can help identify cyclical patterns that may affect crop yield.Based on insights gained from EDA, engineer new features that might be predictive of crop yield. This could involve creating lag features from time series data, deriving new variables from existing ones, or incorporating external data sources.Consider domain knowledge and expert advice when engineering features, as certain factors may have a significant impact on crop yield but may not be immediately apparent from the data. Based on the

insights gained from data exploration and analysis, choose appropriate models for crop yield prediction. This could include regression models, time series forecasting models, machine learning algorithms, or a combination of these.Consider the complexity of your data and the interpretability of the models when making your selection. Based on insights gained from data exploration, engineer new features that capture relevant information and enhance predictive performance. This may include aggregating or transforming existing features, creating lag variables, or incorporating external datasets.Formulate hypotheses about the factors influencing crop yield and test them using statistical methods such as t-tests, ANOVA, or regression analysis.Evaluate the significance of relationships between independent variables and the dependent variable to identify key predictors.Utilize interactive visualization tools and dashboards to facilitate exploratory data analysis, allowing users to interactively explore different aspects of the data and gain insights dynamically.

3.4 Algorithms Used

3.4.1 Random Forest algorithm

Random forest model for crop yield prediction offers a powerful tool for agricultural forecasting. Random forest algorithms leverage the collective intelligence of multiple decision trees, each trained on a subset of the data and employing random feature selection. This approach enables the model to capture complex nonlinear relationships between various factors influencing crop yield, including soil properties, weather conditions, crop management practices, and fertilizer application. By integrating diverse datasets such as historical yield data, soil quality assessments, meteorological records, and fertilizer usage, the random forest model can effectively learn patterns and make predictions with high accuracy. Moreover, the model's ability to handle large and heterogeneous

datasets makes it suitable for accommodating the diverse array of factors influencing crop productivity.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Overview of the components

Model Design

Design the architecture of the neural network based on the selected model architecture. This involves specifying the number of layers, the types of layers (e.g., LSTM, CNN, fully connected), the activation functions, and the dropout regularization.Experiment with different network architectures and hyperparameters to find the optimal configuration for your dataset.

Model Training

Train the neural network using the training data. Use backpropagation and optimization algorithms such as stochastic gradient descent (SGD), Adam, or RMSprop to update the network parameters iteratively.Monitor the training process by tracking metrics such as loss and validation accuracy to prevent overfitting and ensure convergence.

Model Evaluation

Evaluate the trained model's performance using the validation set. Measure performance using appropriate evaluation metrics such as mean squared error, mean absolute error, or coefficient of determination.Compare the model's predictions against the actual crop yields to assess its accuracy and generalization ability.

Model Deployment

Deploy the trained neural network as a component in your crop yield prediction system. Implement an interface or API that allows users to input relevant data and receive crop yield predictions as output.Consider scalability, reliability, and security requirements when deploying the model in a production environment.

Monitoring and Maintenance

Monitor the deployed model's performance over time and retrain it periodically with new data to ensure it remains accurate and up-to-date.Continuously collect feedback from users and stakeholders to identify areas for improvement and update the model accordingly.Stay informed about advancements in deep learning research and techniques to incorporate new insights into your crop yield prediction model.

Steps of Random Forest algorithm

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Random Forest Algorithm working principle

Random Sampling

Random Forest starts by randomly selecting samples from the dataset with replacement. This process, known as bootstrapping, creates multiple subsets of the original dataset, each called a bootstrap sample.

Random Feature Selection

For each split in the decision tree, Random Forest selects a random subset of features from the dataset. This helps in reducing correlation between the trees and makes the algorithm more robust to noise and overfitting.

Building Decision Trees

Decision trees are constructed using the bootstrapped samples and randomly selected features. Each tree is grown recursively by selecting the best split at each node based on a chosen criterion (e.g., Gini impurity for classification, mean squared error for regression). This process continues until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples per leaf node.

Voting or Averaging

Once all the decision trees are built, predictions are made for each tree individually. For classification tasks, each tree's prediction is counted as a vote, and the class with the most votes becomes the final prediction (mode). For regression tasks, the predictions from all trees are averaged to obtain the final prediction.

Ensemble Learning

The final prediction from the Random Forest is determined by aggregating the predictions of all the individual trees. This ensemble approach helps to improve the model's generalization ability, reduce variance, and handle noisy or complex datasets effectively.

Tuning Hyperparameters

Random Forest offers several hyperparameters that can be tuned to optimize performance, such as the number of trees in the forest, the maximum depth of each tree, the minimum number of samples required to split a node, and the number of features to consider at each split.

Evaluation

Random Forest models can be evaluated using standard metrics such as accuracy, precision, recall, F1-score for classification tasks, or mean squared error, R-squared for regression tasks. Additionally, techniques like cross-validation can be employd to estimate the model's performance on unseen data.

3.4.2 Linear Regression

Gather historical data on crop yields and relevant predictor variables such as weather conditions soil characteristics and agricultural practices.Preprocess the data by handling missing values, outliers, and scaling numerical features if necessary.Select a subset of predictor variables that are likely to influence crop yield based on domain knowledge and exploratory data analysis. It's important to include features that are known to have a significant impact on crop growth and development.Remove irrelevant or redundant features that may not contribute much to the predictive performance of the model.Split the dataset into training and testing sets. The training set will be used to train the linear regression model, while the testing set will be used to evaluate its performance.Optionally, you can also set aside a validation set for hyperparameter tuning if needed.Train a linear

regression model using the training data. In linear regression, the relationship between the dependent variable and the independent variables (predictors) is assumed to be linear.Fit the model to the training data using a suitable algorithm to estimate the coefficients of the linear equation.Evaluate the performance of the linear regression model using the testing set. Common evaluation metrics for regression tasks include mean squared error (MSE), mean absolute error (MAE), and R-squared (coefficient of determination).Compare the predicted crop yields with the actual values to assess the accuracy and reliability of the model.Interpret the coefficients of the linear regression model to understand the relationship between the predictor variables and crop yield. Positive coefficients indicate a positive effect on crop yield, while negative coefficients indicate a negative effect.Identify the most influential predictors and their respective impact on crop yield, which can provide valuable insights for crop management and decision-making.Validate the linear regression model using cross-validation or bootstrapping techniques to ensure its generalizability to new data.If the model performance is not satisfactory, consider refining the model by including additional features, incorporating polynomial or interaction terms, or trying more advanced regression techniques.Once validated, use the trained linear regression model to make predictions on new or unseen data. Input the relevant predictor variables into the model to obtain predictions of crop yield for future growing seasons.

3.4.2.1 Steps of Linear Regression

STEP 1: Load the data into python. Follow these four steps for each dataset: ...

STEP 2: Make sure your data meet the assumptions.

STEP 3: Perform the linear regression analysis.

STEP 4: Check for homoscedasticity.

STEP 5: Visualize the results with a graph.

STEP 6: Report your results.

Linear Regression algorithm working principle

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. In the context of climate prediction, linear regression can be used to predict climate variables (e.g., temperature, precipitation) based on historical data and other relevant factors (e.g., greenhouse gas concentrations, solar radiation). Here's an overview of how linear regression works .

Data Collection

Collect historical climate data, including the dependent variable (e.g., temperature) and independent variables (e.g., greenhouse gas concentrations, solar radiation), over a significant period.

Data Preprocessing

Clean the data to remove noise and outliers. Handle missing values by imputation or removal. Ensure the data is in a suitable format for linear regression analysis.

Feature Selection

Select relevant independent variables that are likely to have a significant impact on the dependent variable. This can be done based on domain knowledge or using techniques like correlation analysis.

Prediction

Once the model is trained and evaluated, use it to make predictions on new data. Monitor the model's performance over time and retrain it periodically to maintain its accuracy.

Uncertainty Estimation

Linear regression provides point estimates, but it's important to also estimate the uncertainty around these estimates. This can be done using techniques like bootstrapping or by calculating confidence intervals.

Post-processing

Post-process the model outputs if necessary to derive useful information for decision-making. This could involve aggregating predictions over regions or time periods, or translating them into actionable insights.

Communication

Communicate the linear regression model results and uncertainties effectively to stakeholders, policymakers, and the public to support informed decision-making.

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Data Set

The dataset being used for our prediction models comprises of crop yield records of the city in focus collected over a period of time using various different Factors such as temperature, precipitation, humidity, and sunlight play crucial roles in determining crop yields. Historical weather data as well as forecasts can be used to predict how these variables will affect crop growth

The data enclosed in our dataset is classified into the following Categories:-

i) Temperature

ii) precipitation

iii) sunlight

Different crops have specific temperature requirements for optimal growth. For instance, warm-season crops like maize and soybeans thrive in temperatures ranging from 21°C to 30°C, while cool-season crops like wheat and barley prefer temperatures between 15°C and 24°C. Adequate water is essential for crop growth. Precipitation provides moisture necessary for seed germination, plant growth, and development. Excessive rainfall or poor drainage can result in waterlogging, depriving plant roots of oxygen and causing root rot, nutrient leaching, and reduced yields. The distribution of rainfall throughout the growing season is critical. Uneven or erratic rainfall patterns can affect crop development and yield variability. Sunlight is essential for photosynthesis, the process by which

plants convert light energy into chemical energy, fueling growth Some crops are sensitive to day length or photoperiod, which influences flowering, fruiting, and dormancy. Day length can affect the timing of developmental stages and ultimately impact Intensity and duration of sunlight influence crop canopy development, leaf area expansion, and photosynthetic activity. Adequate solar radiation promotes higher yields by maximizing biomass production and nutrient assimilation. Competition for sunlight among plants can occur in dense stands or canopies, leading to shading and reduced photosynthetic efficiency in lower leaves, affecting overall yield potential.

4.2 Accuracy

Accurate crop yield prediction relies heavily on the availability of high-quality data. This includes historical yield data, weather records, soil information, crop management practices, and satellite imagery. The more comprehensive and detailed the data, the better the accuracy of predictions is likely to be.Advanced statistical and machine learning models can offer higher accuracy compared to simpler models. These models can capture complex relationships between various factors affecting crop yield and make more nuanced predictions. However, the complexity of models also requires careful validation and testing to ensure reliability.The temporal and spatial resolution of data can impact prediction accuracy. High-resolution data, such as daily weather observations and fine-scale satellite imagery, can capture smaller-scale variations in environmental conditions and improve the accuracy of predictions.Predictive models need to be calibrated and validated using independent datasets to assess their accuracy. This involves dividing the available data into training and testing sets, where the training set is used to train the model, and the testing set is used to evaluate its performance. Cross-validation techniques can also help assess model robustness.Despite efforts to improve accuracy, there will always be inherent uncertainty and variability in

crop yield predictions. Weather forecasts, in particular, are subject to uncertainty, which can propagate into yield predictions. It's essential to communicate the uncertainty associated with predictions to users and stakeholders to manage expectations effectively. Local factors, such as microclimatic conditions, soil variability, and pest pressure, can significantly influence crop yield but may not always be captured in broad-scale predictive models. Integrating expert knowledge and local insights into the modeling process can help improve prediction accuracy, especially at the field or farm level.Crop yield prediction models should be continually refined and updated as new data become available and as advancements in modeling techniques emerge. This iterative process of model improvement can lead to increased accuracy over time.

Accuracy= TP / (TP + FP)

Where:

TP is the number of real positives

FP the number of false positives.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where:

n is the number of observations.

yi is the actual observed value for the i-th data point.

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}}$$

Where:

SSresidual is the sum of squares of residuals, also known as the residual sum of squares (RSS). It measures the difference between the actual values and the predicted values of the dependent variable.

SStotal is the total sum of squares (TSS), which measures the total variance in the dependent variable.

Table Comparison of proposed with other existing model

Techniques	Accuracy	Mean	R2 Score
Random forest (Proposed)	0.987183	87087739.870799	0.987183
Linear Regression	0.074374	6289307139.935912	0.074374
Gradient Boost	0.833331	1132456443.447738	0.833331
KNN	0.365903	4308463826.971797	0.365903
Decision Tree	0.976414	160261119.839164	0.976414
Bagging Regressor	0.987120	87516206.904644	0.987120

4.3 Test Setup

The test process is already in-built in our system. The testing process taking place just after the model is trained. After the completion of the training process, we analyze each data entry in the test set. In order to analyze each entry, we use descriptors to extract features. Now we compare these feature values with the feature values which were initially retained using the train set. The comparison is done according to the Machine Learning model used and finally the output for each entry is received. Since each data entry is already labeled, we can compute accuracy by comparing the predicted value with the received values.

4.5 Precision

High-quality data is fundamental for precise yield predictions. This includes accurate and up-to-date information on weather patterns, soil characteristics, crop genetics, management practices, and historical yield data. Ensuring data integrity, consistency, and completeness enhances the precision of predictive models. Model Choosing appropriate modeling techniques tailored to the specific characteristics of the target crop and the agroecological context is crucial for precision. Advanced statistical methods, machine learning algorithms, and crop simulation models can be employed to develop predictive models that capture the complex interactions among various factors influencing crop yields. Identifying relevant predictors and fine-tuning model parameters play a significant role in enhancing precision. Feature selection techniques help identify the most informative variables, while parameter optimization ensures that the model captures the underlying relationships accurately.Validation and Calibration: Rigorous validation procedures are essential to assess the performance of predictive models and verify their precision. Cross-validation, holdout validation, and independent validation using unseen data are common techniques for evaluating model accuracy. Calibration techniques can further refine model predictions by adjusting for biases and improving reliability Precision can be enhanced by incorporating spatial and temporal variability into predictive models. Fine-scale spatial data and high-frequency temporal data enable more accurate characterization of local conditions and yield potential.Acknowledging and quantifying uncertainties associated with yield predictions is critical for decision-making under uncertainty. Uncertainty estimation techniques, such as probabilistic modeling, Monte Carlo simulations, and sensitivity analysis, help assess the robustness of predictions and inform risk management strategies. Combining information from diverse data sources, including satellite imagery, weather forecasts, sensor data, and expert knowledge,

can improve prediction precision by capturing complementary insights and reducing uncertainties. Precision in crop yield prediction is an iterative process that requires continuous monitoring, evaluation, and refinement of predictive models. Incorporating feedback from stakeholders, updating models with new data, and leveraging advances in technology and methodologies contribute to ongoing improvement in precision.

4.6 Experiment Result

The results of the implementation of the project are demonstrated below.

	Model	Accuracy	MSE	R2_score
0	Linear Regression	0.074374	6289307139.935912	0.074374
1	Random Forest	0.987183	87087739.870799	0.987183
2	Gradient Boost	0.833331	1132456443.447738	0.833331
3	XGBoost	0.976865	157196688.414569	0.976865
4	KNN	0.365903	4308463826.971797	0.365903
5	Decision Tree	0.976414	160261119.839164	0.976414
6	Bagging Regressor	0.987120	87516206.904644	0.987120

Table 4.6 Experiment Result

Above factor mentioned are the differentiating factors which influence the overall funtionallity of an algorithm based for the crop yield prediction. The previously observed algorithm has accuracy around 0.074374 which is better with the MSE 623456071.935912 and R2 Score 0.074374, whereas our proposed model has accuracy around 0.987183 , MSE of 87087739.870789 and R2 score of 0.987183 which is more efficient and accurate which provides us with better and faster results.

Comparison of Proposed with other existing algorithms

i. Random Forest (Proposed)
ii. Linear Regression
iii. Gradient Boost
iv. Decision Tree
v. Bagging Regressor
vi. KNN

Linear Regression

Linear regression can indeed be used for crop yield prediction, especially when there's a linear relationship between the features (such as weather conditions, soil properties, etc.) and the crop yield.Gather data on various factors that influence crop yield. This could include weather data (temperature, rainfall, humidity, etc.), soil properties (pH, fertility, etc.), crop type, agricultural practices, etc. Ensure that you have historical data on crop yields corresponding to these factors.Data Preprocessing: Clean the data by handling missing values, removing outliers, and converting categorical variables into numerical representations if necessary. Also, split the data into training and testing sets for model evaluation.

Identify the features (independent variables) that are most likely to influence crop yield. This could be done through domain knowledge or using techniques like correlation analysis.Model Training: Use the training data to fit a linear regression model. This involves finding the coefficients that minimize the error between the predicted and actual crop yields. This can be done using various optimization algorithms like ordinary least squares (OLS) or gradient descent.

Random Forest

Train the Random Forest model using the prepared dataset. The model will learn the relationships between the input variables (e.g., weather, soil conditions,

agricultural practices) and the target variable (crop yield).Evaluate the trained model's performance using appropriate metrics such as mean squared error (MSE), root mean squared error (RMSE), coefficient of determination (R-squared), etc. This step helps assess how well the model generalizes to unseen dataOnce the model is trained and evaluated, it can be used to predict crop yields for new or future instances based on the input variables.Depending on the performance of the model, fine-tuning parameters such as the number of trees, maximum depth of trees, and minimum number of samples required to split a node can be adjusted to optimize performance further.

Gradient Boost

Gradient Boosting is a powerful machine learning algorithm used for predictive modeling tasks, including crop yield prediction Gradient Boosting is chosen as the modeling algorithm due to its effectiveness in handling complex relationships in the data and its ability to handle both numerical and categorical features.Train the Gradient Boosting model using the training data. During training, the model iteratively fits weak learners (typically decision trees) to the residuals of the previous iteration, gradually reducing the errors.Gradient Boosting has several hyperparameters that can be tuned to optimize performance, such as the number of trees (or iterations), the learning rate, maximum depth of trees, and minimum samples per leaf. Hyperparameter tuning can be done using techniques like grid search or randomized search.Evaluate the trained model using the testing set. Common evaluation metrics for regression tasks like crop yield prediction include mean absolute error (MAE), mean squared error (MSE), and R-squared.Once the model is trained and evaluated, it can be used to make predictions on new or unseen data. These predictions can help farmers make informed decisions about crop management practices, such as irrigation scheduling, fertilizer application, and pest control.

Decision Tree

Decision tree algorithms are advantageous for crop yield prediction because they can handle both numerical and categorical data, they're interpretable (you can visualize the decision-making process), and they're relatively easy to implement and understand. It's crucial to validate the model's predictions against actual crop yields to assess its accuracy. If necessary, the model can be refined by tweaking parameters, selecting different features, or using more advanced techniques. You evaluate the performance of the decision tree model using metrics such as accuracy, precision, recall, or Mean Absolute Error (MAE) depending on whether it's a classification or regression problem. Common evaluation metrics for regression tasks like crop yield prediction include mean absolute error (MAE), mean squared error (MSE), and R-squared.Once the model is trained and evaluated, it can be used to make predictions on new or unseen data. These predictions can help farmers make informed decisions about crop management practices, such as irrigation scheduling, fertilizer application, and pest control.

Bagging Regression

Bagging (Bootstrap Aggregating) is a popular ensemble learning technique that can be applied to regression problems, including crop yield prediction. Bagging involves creating multiple models using different subsets of the training data and then combining their predictions to obtain a more Randomly sample subsets of the training data with replacement Train a base regression model on each bootstrap sample.Aggregate the predictions of all base models to make the final prediction. Once the bagging ensemble model is trained, you can use it to predict crop yields for new data instances. Aggregate the predictions from all base models to obtain the final prediction. For regression problems, the aggregation can be done by taking the average of the individual predictions

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

On implementing two machine learning models, Convolutional Neural Networks (CNN) and K-nearest Neighbors (KNN) on the disease detection of tomato leaves from the plant village dataset and also evaluating the aforementioned model using the following metrics: Accuracy, Precision, Recall and F1-Score, the study shows that CNN model performs better than the KNN model in the plant disease detection of tomato leaves by outperforming the KNN model is all of the four evaluation metrics. The study also makes use of the XAI technique Local Interpretable Model-agnostic Explanations (LIME) in order to provide explain ability to the predictions made by the models. With the execution of a user study, this study is able to get feedback from farmers on if they trust the aforementioned AI and XAI models. The results from the user study indicate that the farmers find the predictions and explanations from AI and XAI models inadequate and therefore do not trust the implemented tools for the detection of plant diseases. However, through additional feedback the farmers highlight areas that could possibly help improve and trust the AI and XAI models.The dataset used in this study makes use of only tomato leaves from the plant village dataset. Due to the lack of enough Random Access Memory (RAM) storage the study had to be limited to only 10,000 images. In the future, it would be great to test the implementation of both the CNN and KNN model and also use LIME on the whole plant village dataset containing multiple different plants in order to bring detection and explain ability to a wide variety of plants.Another work that this study would like to pursue in the future is to provide a comparative study on different XAI techniques and implement a user study in order to find out which

XAI technique provides the best explain ability, transparency and interpretability. With the addition of data on Volatile organic compounds, soil typezzenvironmental conditions and time of the month as mentioned by farmers through feedback from the user study, the user trust of the detection tool is expected to grow a little higher. As discussed earlier in the use case of this study, a working application that is capable of taking pictures of plants and detecting plant diseases in real-time is the ideal goal and will prove to be of great use to the farmers and botany enthusiasts.

5.2 Future Work

Imagery, weather data, soil composition maps, and historical crop yield records can enhance the predictive power of CNN models. Integrating multi-modal data streams requires advanced fusion techniques and model architectures capable of handling heterogeneous inputs.Temporal Analysis: Extending CNN models to analyze temporal trends in remote sensing data enables the capture of seasonal variations, phenological stages, and long-term changes in crop growth patterns. Time-series CNN architectures can effectively model temporal dependencies and provide insights into crop yield dynamics over time.Developing CNN models for crop type classification facilitates the identification of different crop species within agricultural fields. This information is essential for tailoring yield prediction models to specific crops and optimizing agricultural management practices accordingly. CNNs can be trained to predict crop yield at a fine spatial resolution, enabling the generation of detailed yield maps that highlight within-field variability. High-resolution satellite imagery and drone-based data acquisition techniques offer the spatial granularity needed for accurate yield estimation at the field level.Transfer Learning and Domain Adaptation: Leveraging transfer learning techniques, pre-trained CNN models can be fine-tuned on agricultural datasets to expedite model training and improve generalization performance. Domain adaptation methods allow CNN models

trained on data from one geographic region or growing season to be adapted to different regions or time periods with minimal labeled data.Uncertainty Estimation: Integrating uncertainty estimation techniques into CNN-based crop yield prediction models provides decision-makers with valuable information about prediction reliability and confidence intervals. Bayesian CNNs, dropout-based uncertainty estimation, or ensemble methods can quantify prediction uncertainty and assist in risk assessment and decision-making under uncertainty.Developing interactive visualization tools that leverage CNN predictions and spatial data enables stakeholders to explore and interpret crop yield estimates effectively. Interactive dashboards, web-based interfaces, and mobile a make informed decisions based on real-time yield predictions and spatial analytics.

ANNEXURE

6.1 Sample code

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import geopandas as gpd
import plotly.express as px

class colorss:

yellows=['#ffffd4','#fee391','#fec44f','#fe9929','#d95f0e','#993404','#a70000','#ff5252','#ff7b7b','#ffbaba']

greens=['#ffffd4','#fee391','#fec44f','#fe9929','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32']
cmaps=['flare','icefire','bwr_r','Accent','Spectral','RdGy','afmhot_r','afmhot','inferno','seismic','vlag','vlag_r']
from sklearn.preprocessing import LabelEncoder
categorical_columns = datacorr.select_dtypes(include=['object']).columns.tolist()
label_encoder = LabelEncoder()
for column in categorical_columns:
    datacorr[column] = label_encoder.fit_transform(datacorr[column])

geojson_url = "https://raw.githubusercontent.com/nvkelso/natural-earth-vector/master/geojson/ne_110m_admin_0_countries.geojson"
```

```
data = gpd.read_file(geojson_url)


merged_data = data.merge(df, left_on='NAME', right_on='Area', how='left')
# fig, ax = plt.subplots(figsize=(15, 10))
merged_data.plot( column='hg/ha_yield', cmap='Greens_r', linewidth=0.8, edgecolor='0.8')
# plt.title("Countries")
plt.show()
palette = sns.color_palette('tab20', 21,as_cmap=True)
num_plots = 7
areas_per_plot = 10



# Get unique areas
unique_areas = sorted(df['Area'].unique())


# Split into chunks
area_chunks = [unique_areas[i:i+areas_per_plot] for i in range(0, len(unique_areas), areas_per_plot)]
area_chunks[-2] = unique_areas[-11:]
fig, axs = plt.subplots(ncols=num_plots, figsize=(30, 10))
j=0
for i, ax in enumerate(axs):


    plot_df = df[df['Area'].isin(area_chunks[i])]
    for i, area in enumerate(plot_df['Area'].unique()):
      data = plot_df[plot_df['Area'] == area]
      ax.hist(data['hg/ha_yield'], facecolor=palette(i), label=area)
```

```
    ax.legend()
    j+=1

plt.show()
dk=df.groupby(['Area','Item'])['hg/ha_yield'].mean().to_frame()
dk.sort_values(by=['hg/ha_yield'],ascending=False)
for i in range(0,7):
palette = sns.color_palette('tab20', 21,as_cmap=True)
num_plots = 7
areas_per_plot = 10

# Get unique areas
unique_areas = sorted(df['Area'].unique())

# Split into chunks
area_chunks = [unique_areas[i:i+areas_per_plot] for i in range(0,
len(unique_areas), areas_per_plot)]
area_chunks[-2] = unique_areas[-11:]
fig, axs = plt.subplots(ncols=num_plots, figsize=(30, 10))
j=0
for i, ax in enumerate(axs):

    plot_df = df[df['Area'].isin(area_chunks[i])]
    for i, area in enumerate(plot_df['Area'].unique()):
        data = plot_df[plot_df['Area'] == area]
        ax.hist(data['hg/ha_yield'], facecolor=palette(i), label=area)
```

```
    ax.legend()
    j+=1


plt.show()
    plot_df = df[df['Area'].isin(area_chunks[i])]


plot_df.groupby(['Area'])['average_rain_fall_mm_per_year'].mean().plot(kind='
bar',rot=0,color=colorss.greens)
    plt.xticks(rotation=90)
    plt.show()
for i in range(0,7):
    plot_df = df[df['Area'].isin(area_chunks[i])]


plot_df.groupby(['Area'])['pesticides_tonnes'].mean().plot(kind='bar',rot=0,color
=colorss.yellows)
    plt.xticks(rotation=90)
    plt.show()
for i in range(0,7):
    plot_df = df[df['Area'].isin(area_chunks[i])]
    plot_df.groupby('Area')[['pesticides_tonnes',
'hg/ha_yield']].mean().plot(kind='bar',rot=0,color=colorss.yellows[-6:])
    plt.xticks(rotation=90)
    plt.show()
px.scatter(df, x='hg/ha_yield',
y='pesticides_tonnes',color="Area",color_discrete_sequence=colorss.greens)
num_plots = 7
areas_per_plot = 10
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
```

```
from sklearn.metrics import mean_squared_error, r2_score,
mean_absolute_error, mean_absolute_percentage_error

from xgboost import XGBRegressor

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.model_selection import cross_val_score

from sklearn.model_selection import KFold

from sklearn.neighbors import KNeighborsRegressor

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import BaggingRegressor

X, y = datacorr.drop(labels='hg/ha_yield', axis=1), datacorr['hg/ha_yield']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)



# Get unique areas

unique_areas = sorted(df['Area'].unique())


# Split into chunks

area_chunks = [unique_areas[i:i+areas_per_plot] for i in range(0,
len(unique_areas), areas_per_plot)]

area_chunks[-2] = unique_areas[-11:]

fig, axs = plt.subplots(ncols=num_plots)

j=0

for i, ax in enumerate(axs):


    plot_df = df[df['Area'].isin(area_chunks[i])]
```

```
    ax = px.scatter(plot_df, x='hg/ha_yield',
y='pesticides_tonnes',color="Area",color_discrete_sequence=colorss.greens)
    j+=1


    ax.show()
plt.clf()
sns.barplot(data=df, x = df.Item, y = df['pesticides_tonnes'],palette='BrBG')
plt.xticks(rotation=90)
plt.show()
a4_dims = (16.7, 8.27)

fig, ax = plt.subplots(figsize=a4_dims)
sns.boxplot(x="Item",y="hg/ha_yield",palette="BrBG",data=df,ax=ax)
sns.scatterplot(x = 'Item', y = 'avg_temp', data = df,size=10,color='y')
plt.xticks(rotation=90);
grouped = df.groupby('Item')

best_areas = []

for item, group in grouped:
    max_production_row = group[group['hg/ha_yield'] ==
group['hg/ha_yield'].max()]

    area = max_production_row['Area'].values[0]
    production = max_production_row['hg/ha_yield'].values[0]

    best_areas.append({'Item': item, 'Area': area, 'hg/ha_yield': production})
```

```
best_areas_df = pd.DataFrame(best_areas)


best_areas_df

ax=sns.barplot(data=best_areas_df,x='hg/ha_yield',y='Area',hue='Item',palette=colorss.yellows)

ax.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

plt.show()

fig, axes = plt.subplots(4, 1, figsize=(18, 22))


sns.scatterplot(x = "pesticides_tonnes", y = "hg/ha_yield", hue = "Item", data = df, ax=axes[0], legend = True,palette='Spectral')

axes[0].tick_params(axis='x', rotation=45)

axes[0].set_ylabel('Average Yield')

axes[0].legend(bbox_to_anchor=(1.05, 1), loc='upper left')


sns.scatterplot(x = "average_rain_fall_mm_per_year", y = "hg/ha_yield", hue = "Item", data = df, ax=axes[1], legend = True,palette='Spectral')

axes[1].tick_params(axis='x', rotation=45)

axes[1].set_ylabel('Average Yield')

axes[1].legend(bbox_to_anchor=(1.05, 1), loc='upper left')


sns.scatterplot(x = "avg_temp", y = "hg/ha_yield", hue = "Item", data = df, ax=axes[2], legend = True,palette='Spectral')

axes[2].tick_params(axis='x', rotation=45)

axes[2].set_ylabel('Average Yield')

axes[2].legend(bbox_to_anchor=(1.05, 1), loc='upper left')


sns.lineplot(x = "Year", y = "hg/ha_yield", hue = "Item", data = df, ax=axes[3], legend = True,palette='Spectral')
```

```
axes[3].tick_params(axis='x', rotation=45)
axes[3].set_ylabel('Average Yield')
axes[3].legend(bbox_to_anchor=(1.05, 1), loc='upper left')

plt.tight_layout()
plt.show()
def change_of_years(data, template='seaborn'):
    col = data.columns[3:].tolist()
    for i in col:
        sns.lineplot(data.groupby(['Year'])[i].mean(),color='brown')
        plt.title=f'Effect of Years on the {i}'



        plt.show()
        yield()
next(yplot);
next(yplot);
next(yplot);
df_Egypt = df.loc[df['Area'] == 'Egypt']
yplot = change_of_years(df_Egypt)
next(yplot);
next(yplot);
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score,
mean_absolute_error, mean_absolute_percentage_error
from xgboost import XGBRegressor
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import BaggingRegressor
X, y = datacorr.drop(labels='hg/ha_yield', axis=1), datacorr['hg/ha_yield']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
for name, model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = model.score(X_test, y_test)
    MSE = mean_squared_error(y_test, y_pred)
    R2_score = r2_score(y_test, y_pred)
    results.append((name, accuracy, MSE, R2_score))
    acc = (model.score(X_train , y_train)*100)
    print(f'The accuracy of the {name} Model Train is {acc:.2f}')
    acc =(model.score(X_test , y_test)*100)
    print(f'The accuracy of the  {name} Model Test is {acc:.2f}')
    plt.scatter(y_test, y_pred,s=10,color='#9B673C')
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
#     plt.title(f' {name} Evaluation')
    plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='green',
linewidth = 4)
    plt.show()
```

```
dff = pd.DataFrame(results, columns=['Model', 'Accuracy', 'MSE', 'R2_score'])

df_styled_best = dff.style.highlight_max(subset=['Accuracy','R2_score'],
color='green').highlight_min(subset=['MSE'],
color='green').highlight_max(subset=['MSE'],
color='red').highlight_min(subset=['Accuracy','R2_score'], color='red')


# df_styled_worst = dff.style.highlight_max(subset=['MSE'],
color='red').highlight_min(subset=['Accuracy','R2_score'], color='red')

class colorss:


yellows=['#ffffd4','#fee391','#fec44f','#fe9929','#d95f0e','#993404','#a70000','#ff
5252','#ff7b7b','#ffbaba']


greens=['#ffffd4','#fee391','#fec44f','#fe9929','#d9f0a3','#addd8e','#78c679','#41a
b5d','#238443','#005a32']

cmaps=['flare','icefire','bwr_r','Accent','Spectral','RdGy','afmhot_r','afmhot','infer
no','seismic','vlag','vlag_r']

from sklearn.preprocessing import LabelEncoder

categorical_columns = datacorr.select_dtypes(include=['object']).columns.tolist()

label_encoder = LabelEncoder()

for column in categorical_columns:

    datacorr[column] = label_encoder.fit_transform(datacorr[column])


geojson_url = "https://raw.githubusercontent.com/nvkelso/natural-earth-
vector/master/geojson/ne_110m_admin_0_countries.geojson"

data = gpd.read_file(geojson_url)


merged_data = data.merge(df, left_on='NAME', right_on='Area', how='left')

# fig, ax = plt.subplots(figsize=(15, 10))
```

```
display(df_styled_best)
# display(df_styled_worst)
results = []

models = [
    ('Linear Regression', LinearRegression()),
    ('Random Forest', RandomForestRegressor(random_state=42)),
    ('Gradient Boost', GradientBoostingRegressor(n_estimators=100,
 learning_rate=0.1, max_depth=3,random_state=42)),
    ('XGBoost', XGBRegressor(random_state=42)),
     ('KNN',KNeighborsRegressor(n_neighbors=5)),
     ('Decision Tree',DecisionTreeRegressor(random_state=42)),
    ('Bagging Regressor',BaggingRegressor(n_estimators=150,
random_state=42))
          ]
for item, group in grouped:
    max_production_row = group[group['hg/ha_yield'] ==
group['hg/ha_yield'].max()]

    area = max_production_row['Area'].values[0]
     production = max_production_row['hg/ha_yield'].values[0]

    best_areas.append({'Item': item, 'Area': area, 'hg/ha_yield': production})

best_areas_df = pd.DataFrame(best_areas)

best_areas_df
```

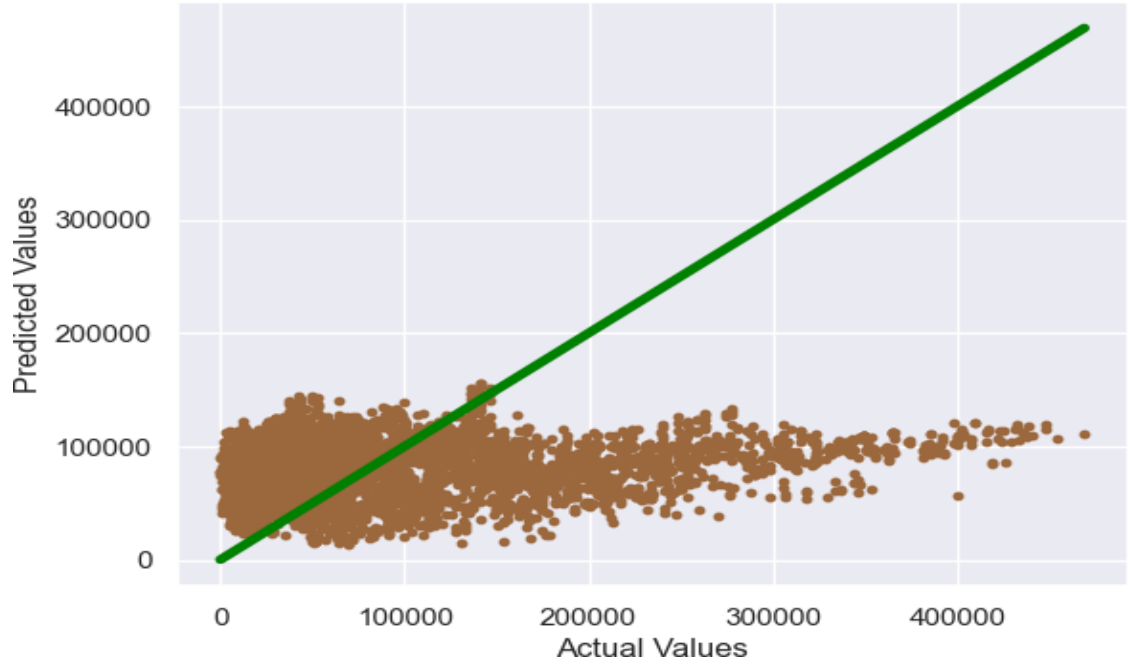
```
ax=sns.barplot(data=best_areas_df,x='hg/ha_yield',y='Area',hue='Item',palette=colorss.yellows)

ax.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

plt.show()

fig, axes = plt.subplots(4, 1, figsize=(18, 22))


sns.scatterplot(x = "pesticides_tonnes", y = "hg/ha_yield", hue = "Item", data = df, ax=axes[0], legend = True,palette='Spectral')

axes[0].tick_params(axis='x', rotation=45)

axes[0].set_ylabel('Average Yield')

axes[0].legend(bbox_to_anchor=(1.05, 1), loc='upper left')


sns.scatterplot(x = "average_rain_fall_mm_per_year", y = "hg/ha_yield", hue = "Item", data = df, ax=axes[1], legend = True,palette='Spectral')

axes[1].tick_params(axis='x', rotation=45)

axes[1].set_ylabel('Average Yield')

axes[1].legend(bbox_to_anchor=(1.05, 1), loc='upper left')



for name, model in models:

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    accuracy = model.score(X_test, y_test)

    MSE = mean_squared_error(y_test, y_pred)

    MAE = mean_absolute_error(y_test, y_pred)

    MAPE = mean_absolute_percentage_error(y_test, y_pred)

    R2_score = r2_score(y_test, y_pred)

    results.append((name, accuracy, MSE, MAE, MAPE, R2_score))
```

```
    print(name)
    num_folds = 5
    kf = KFold(n_splits=num_folds, shuffle=True)
    scores = cross_val_score(model, X, y, cv=kf)

    for fold, score in enumerate(scores):
        print(f"Fold {fold+1}: {score}")

    mean_score = np.mean(scores)
    print(f"Mean Score: {mean_score}")
    print('-'*30)



df = pd.DataFrame(results, columns=['Model', 'Accuracy', 'MSE', 'MAE',
'MAPE', 'R2_score'])
df_styled_best =df.style.highlight_max(subset=['Accuracy','R2_score'],
color='lightblue').highlight_min(subset=['MSE','MAE','MAPE'],
```

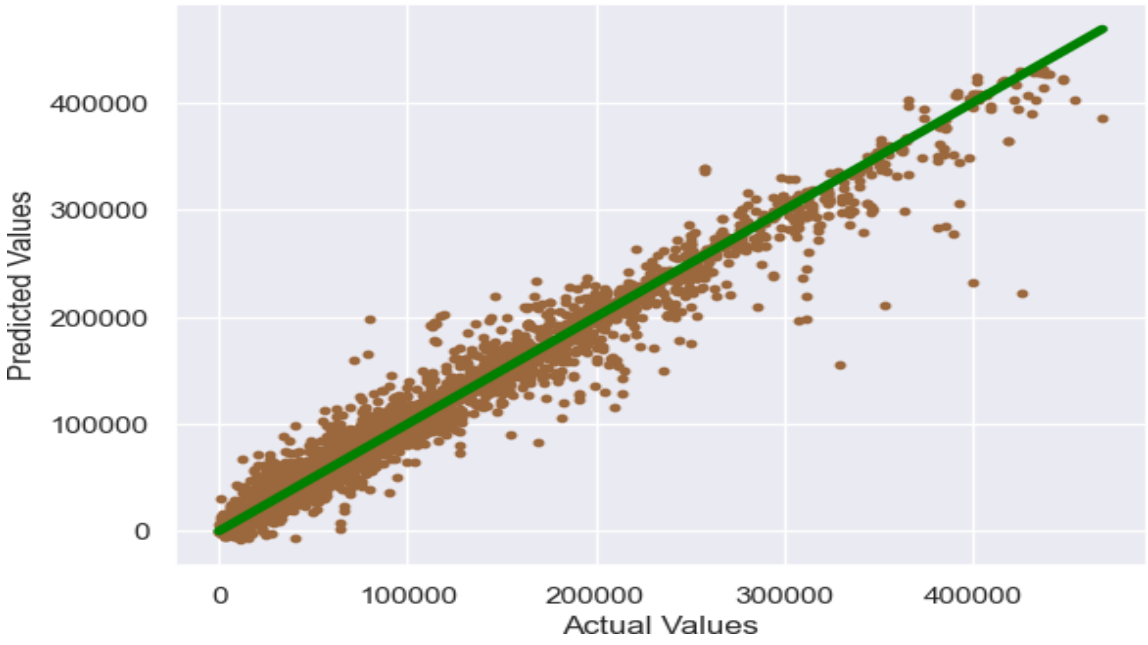
APPENDIX

SCREENSHOTS

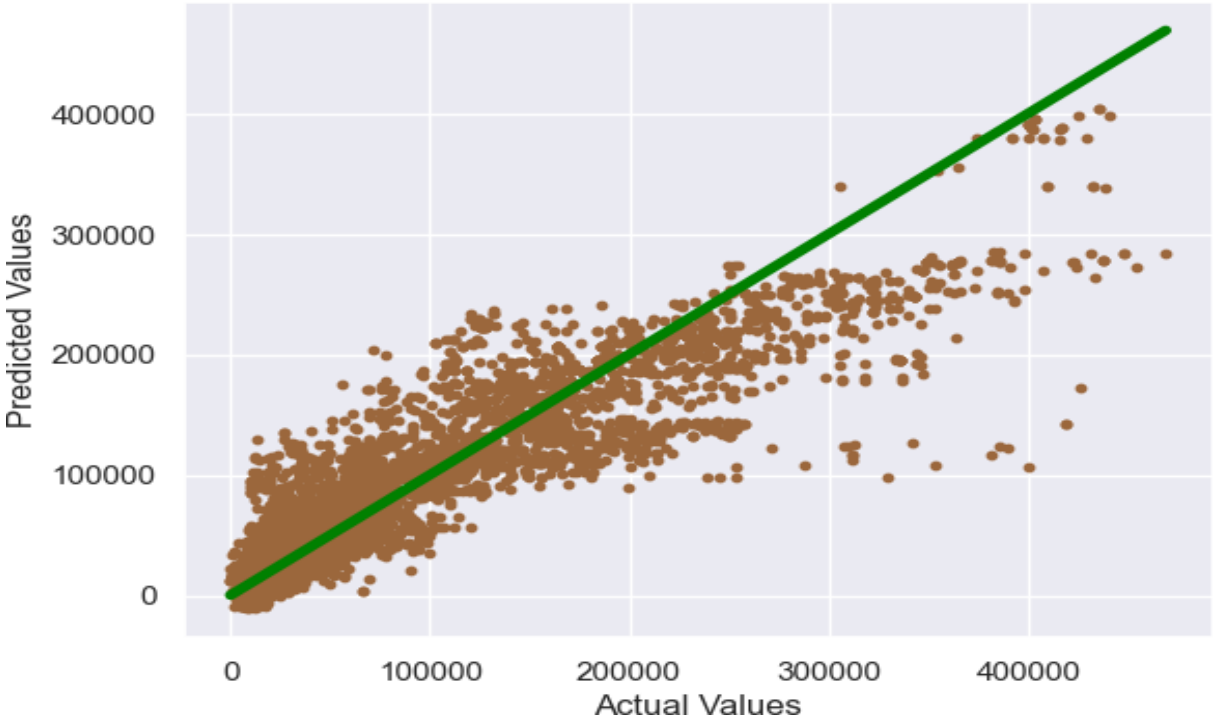
The accuracy of the Linear Regression Model Train is 7.47
The accuracy of the Linear Regression Model Train is 7.77



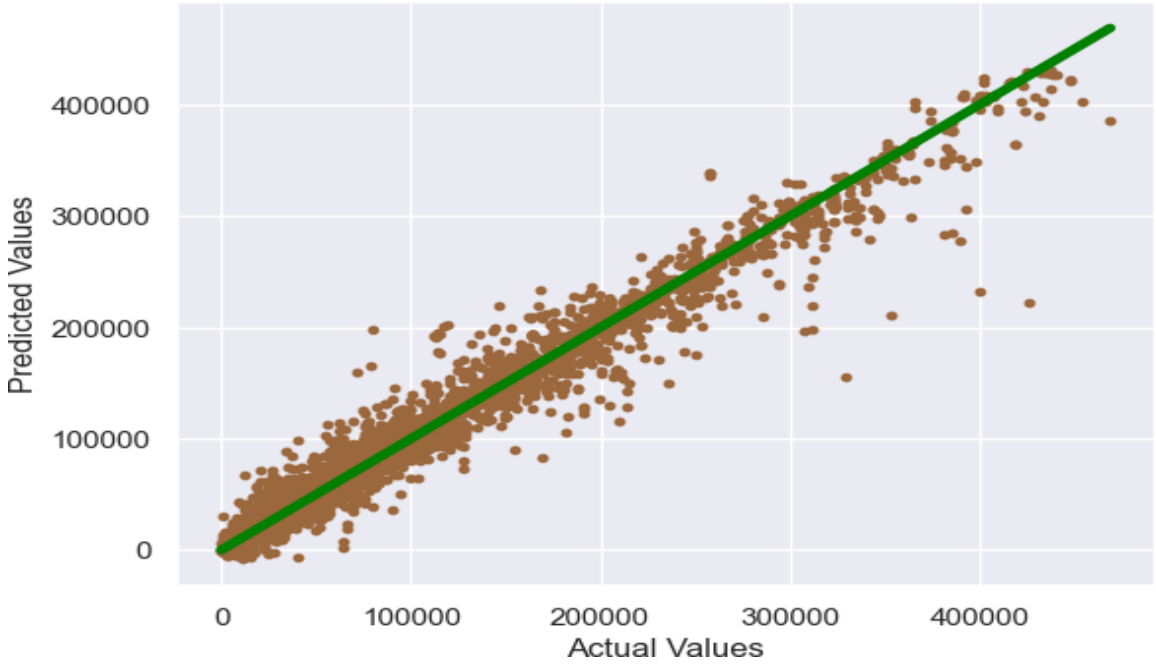
The accuracy of the Random Forest Model Train is 99.82
The accuracy of the Random Forest Model Train is 98.72



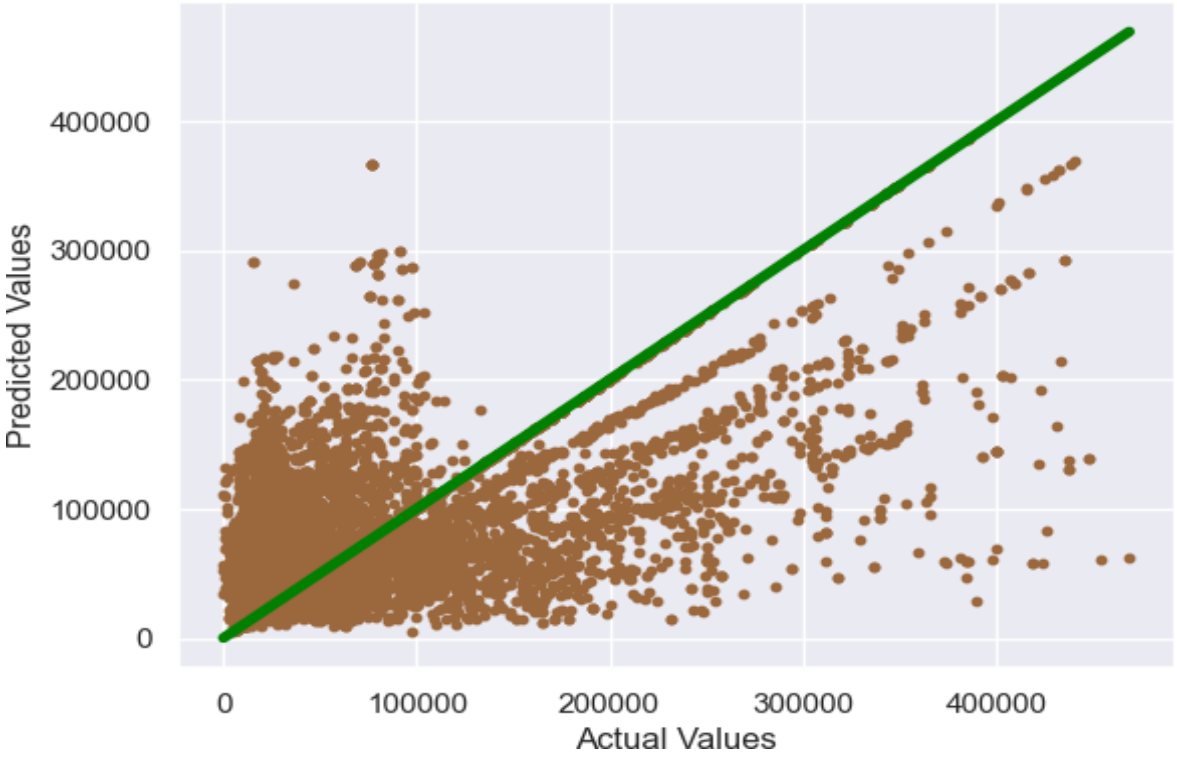
The accuracy of the Gradient Boost Model Train is 84.83
The accuracy of the Gradient Boost Model Train is 83.33



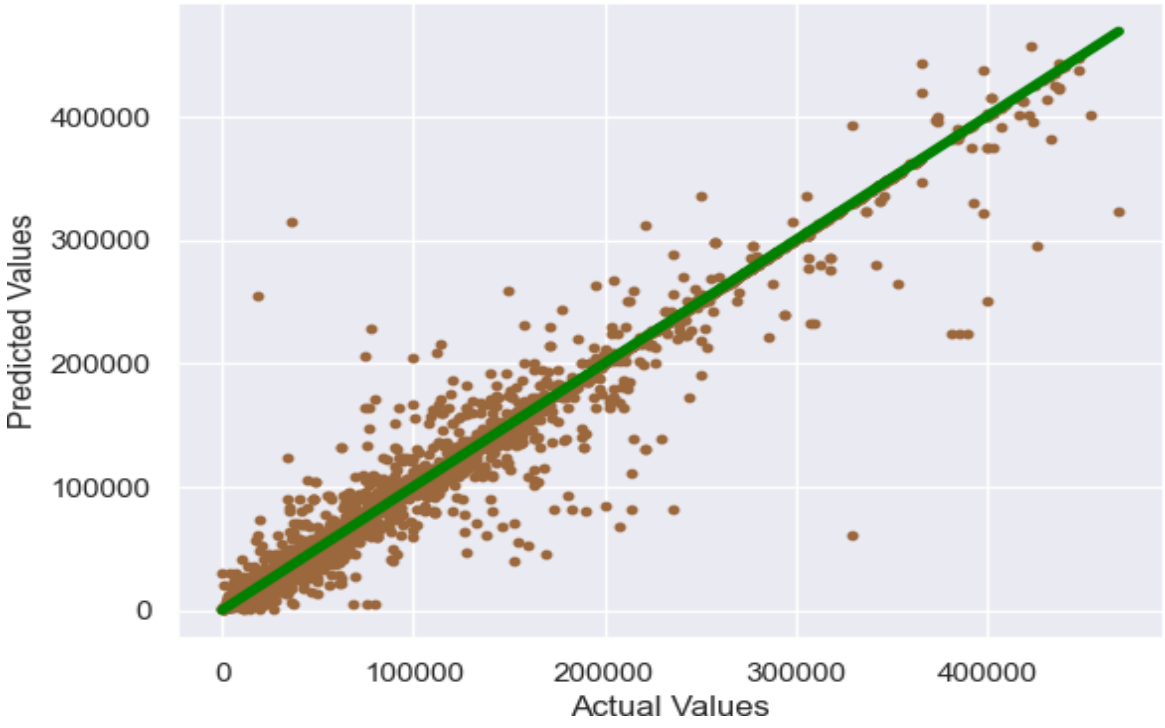
The accuracy of the XGBoost Model Train is 98.90
The accuracy of the XGBoost Model Train is 97.69



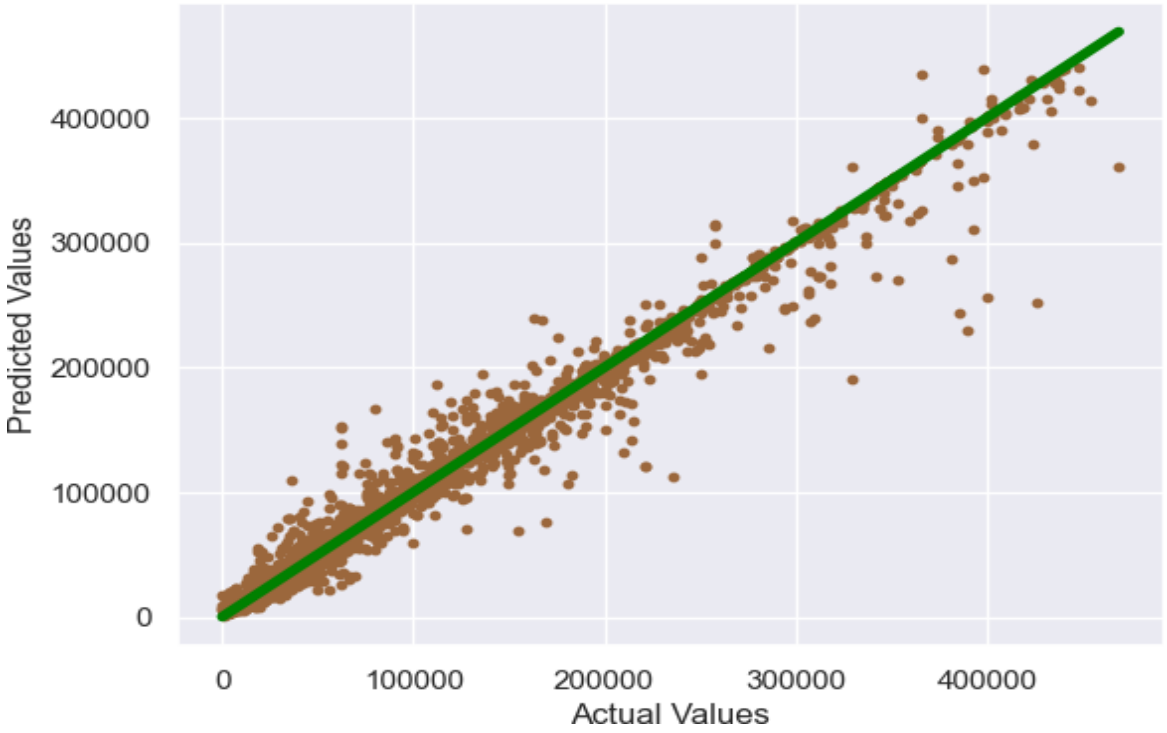
The accuracy of the KNN Model Train is 64.13
The accuracy of the KNN Model Train is 36.59



The accuracy of the Decision Tree Model Train is 100.00
The accuracy of the Decision Tree Model Train is 97.64



The accuracy of the Bagging Regressor Model Train is 99.82
The accuracy of the Bagging Regressor Model Train is 98.71



REFERENCES

[1] Behbood. J. Lu, V., P. Hao, H. Zuo, S. Xue, and G. Zhang, ‘‘Transfer learning using computational intelligence: A survey,’’ Knowl.-Based Syst., vol. 80, pp. 14–23, May 2019.

[2] García-Pedrero .A.M., C. Gonzalo-Martín, M. F. Lillo-Saavedra,DRodriguéz-Esparragón, and E. Menasalvas, ‘‘Convolutional neural networks for estimating spatially distributed evapotranspiration,’’ *Proc. SPIE*, vol. 10427, pp. 198–206, Oct. 2017.

[3] He.K, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2020, pp. 770–778.

[4] Huang.G, Z. Liu, K. Q. Weinberger, and L. van der Maaten, ‘‘Densely connected convolutional networks,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jun. 2019, pp. 2261–2269.

[5] Jaffer.Lv, X. Zheng, M. Pawlak, W. Mo, and M. Miśkowicz, ‘‘Very short-term probabilistic wind power prediction using sparse machine learning and nonparametric density estimation algorithms,’’ Renew. Energy, vol. 177, pp. 181–192, Nov. 2021, doi: 10.1016/j.renene.2021.05.123

[6] JamalD. Domingo, J. Gomez-Garcia-Bermejo, and E. Zalama, ‘‘Optimization and improvement of a robotics gaze control system using LSTM networks,’’ Multimedia Tools Appl., vol. 81, no. 3, pp. 3351–3368, 2021, doi: 10.1007/s11042-021-11112-7.

[7] Mohamed Neshat, M. M. Nezhad, E. Abbasnejad, S. Mirjalili, LTjernberg D. A. Garcia, B. Alexander, and M. Wagner, ‘‘A deep learning-basevolutionary model for short-term wind speed forecasting: A case study of the lillgroffshore wind farm,’’ Energy Convers. Manage., vol. 236, May 2021, Art. no. 114002, doi: 10.1016/j.enconman.2021.114002.

[8] Khan and Ullah.S.D., ‘‘A survey of advances in vision-based vehicle re-identification,’’ Comput. Vis. Image Understand., vol. 182, pp. 50–63, May 2019.

[9] Krizhevsky .A, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Lake Tahoe, NV, USA, Dec. 2019, pp. 1097–1105.

[10] Shakila.L, S. Tellmann, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, Y. Matsuda, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the crop yield from the sub-cloud region to the as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[11] Simonyan and Zisserman .k.A, ‘‘Very deep convolutional networks for large-scale image recognition,’’ 2020, arXiv:1409.1556.

[12] Subhan.D, R. Wang, C. Wei, Y. Wang, Y. Zhou, J. Wang, and H. Song, ‘‘The connectivity evaluation among wells in reservoir utilizing machine learning methods,’’ IEEE Access, vol. 8, pp. 47209–47219, 2020, doi: 10.1109/ACCESS.2020.2976910.

[13] Szegedy.C, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ‘‘Going deeper with convolutions,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Boston, MA, USA, Jun. 2020, pp. 1–9.

[14] Rangarajan.C,Y. Matsuda, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,,, and M. Antonita, ‘‘Thermal structure of the crop yield from the sub-cloud region to the as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[15] Trisha.I, S. Tellmann, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, H. Sagawa, S. Limaye, Y. Matsuda, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the crop yield from the sub-cloud region to the as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020,

[16] Yamal.I, Y. Xiang, L. Ali, and Z. Abdul-Rauf, ‘‘A bidirectional LSTM deep learning approach for intrusion detection,’’ Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115524, doi: 10.1016/j.eswa.2021.115524.

[17] Yin.J, A. Wu, and W. S. Zheng, ‘‘Fine-grained person re-identification,’’ Int. J. Comput. Vis., vol. 128, no. 12, pp. 1654–1672, 2020.

[18] Yamie.D, Wang.J, L. Xiao, and T. Fu, ‘‘Short-term wind speed time series based on a hybrid method with multiple objective optimization for non-convex target,’’ Energy, vol. 215, Jan. 2021, Art. no. 119180, doi: 10.1016/j.energy.2020.119180.

[19] Wei.X.-S, Q. Cui, L. Yang, P. Wang, and L. Liu, ‘‘RPC: A large-scale retail product checkout dataset,’’ 2019, arXiv:1901.07249.

[20] Wei .X.-S., Y.-Z. Song, O. M. Aodha, J. Wu, Y. Peng, J. Tang, J. Yang, and S. Belongie, ‘‘Fine-grained image analysis with deep learning: A survey,’’ IEEE Trans. Pattern Anal. Mach. Intell., early access, Nov. 13, 2021, doi: 10.1109/TPAMI.2021.3126648.