HARVEST PLUS: ENHANCING YIELDS THROUGH PREDICTIVE CROP MODELING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

K. KEERTHI	(21UECM0107)	**(20317)**
K. TEJASWI	(21UECM0108)	**(20190)**
SANAGAVARAPU SHALINI	(21UECM0213)	**(20217)**

Under the guidance of
Mr. Anil Kumar Sandrapuri, Associate Director - IT/Cloud Consulting @ Kyndryl (IBM Spinoff)
&
Dr. K. Kishore Kumar,M.E.,Ph,D.,
Assistant Professor (SG)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

HARVEST PLUS: ENHANCING YIELDS THROUGH PREDICTIVE CROP MODELING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

K. KEERTHI	(21UECM0107)	**(20317)**
K. TEJASWI	(21UECM0108)	**(20190)**
SANAGAVARAPU SHALINI	(21UECM0213)	**(20217)**

Under the guidance of
Mr. Anil Kumar Sandrapuri, Associate Director - IT/Cloud Consulting @ Kyndryl (IBM Spinoff)
&
Dr. K. Kishore Kumar,M.E.,Ph,D.,
Assistant Professor (SG)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”HARVEST PLUS: ENHANCING YIELDS THROUGH PREDICTIVE CROP MODELING” by K. KEERTHI (21UECM0107), K. TEJASWI (21UECM0108), SANAGAVARAPU SHALINI (21UECM0213) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Industry Supervisor
Mr. Anil Kumar Sandrapuri
Associate Director - IT
Cloud Consulting
Kyndryl (IBM Spinoff)
May, 2024

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

K. KEERTHI

Date: / /

K. TEJASWI

Date: / /

SANAGAVARAPU SHALINI

Date: / /

APPROVAL SHEET

This project report entitled HARVEST PLUS: ENHANCING YIELDS THROUGH PREDICTIVE CROP MODELING by K. KEERTHI (21UECM0107), K. TEJASWI (21UECM0108), SANAGAVARAPU SHALINI (21UECM0213) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr. K. KISHORE KUMAR,M.E.,Ph,D.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We are highly indebted to our **Industry Supervisor Mr. Anil Kumar Sandrapuri, Associate Director - IT/Cloud Consulting @ Kyndryl (IBM Spinoff))** for his guidance and constant supervision as well as for providing necessary information and support in completing the project

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Dr. K. KISHORE KUMAR, M.E.,Ph,D.,,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms.E.CHANDRALEKHA,M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

K. KEERTHI	**(21UECM0107)**
K. TEJASWI	**(21UECM0108)**
SANAGAVARAPU SHALINI	**(21UECM0213)**

ABSTRACT

In modern agriculture, accurate prediction of crop yield is paramount for ensuring food security, optimizing resource allocation, and facilitating sustainable farming practices. Traditional methods of estimation, reliant on historical data and manual observations, often lack scalability and precision. However, leveraging machine learning techniques presents a promising solution by harnessing vast datasets, advanced algorithms, and computational power. This project endeavours to develop a robust crop yield prediction model using machine learning, incorporating diverse features such as weather data, soil characteristics, and crop management practices to forecast yields across different regions and crop types. By analyzing historical agricultural data encompassing factors like soil quality, weather patterns, and farming practices, predictive models are trained to anticipate yields for future seasons. Random Forest, a powerful machine learning algorithm, is employed to extract patterns from the data, facilitating accurate yield predictions. These predictions empower farmers to make informed decisions on crop selection, planting strategies, resource allocation, and risk management, ultimately enhancing productivity and sustainability in agriculture. By improving the efficiency of farming operations and bolstering resilience against environmental and market fluctuations, machine learning-based crop yield prediction plays a vital role in ensuring the stability and security of the global food supply.

Keywords: Crop Yield Prediction, Historical Crop Data, Machine Learning, Patterns Extraction, Random Forest, Weather Data.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
GD	Gradient Boosting
ML	Machine Learning
MSE	Mean Squared Error
PH	Potential of Hydrogen
RF	Random Forest
RMSE	Root Mean Squared Error
R2 Score	R-Squared Score

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The project focuses to revolutionize agricultural practices by harnessing the power of Machine Learning (ML), specifically employing the Random Forest (RF) algorithm, to predict crop yield. By amalgamating diverse agricultural datasets encompassing factors such as crop varieties, soil attributes, weather patterns, and agricultural techniques, this endeavor seeks to develop a robust predictive model. Through meticulous preprocessing, feature engineering, and model training, the project endeavors to provide accurate forecasts of crop yield, empowering farmers, agricultural researchers, and policymakers to make informed decisions.The algorithm used in the project is RF which is a versatile and powerful ML algorithm commonly used for both classification and regression tasks. It involves combining multiple individual models to improve overall predictive performance. It is built upon the concept of decision trees. A decision tree is a flowchart-like structure where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a continuous value.

RF builds multiple decision trees during the training phase. The number of trees or forests in the RF is a hyperparameter that can be specified by the user. Each tree is trained on a random subset of the training data and a random subset of the features. This randomness helps to decorrelate the individual trees, making them diverse and less prone to overfitting. RF is highly flexible and can handle both numerical and categorical features. It is robust to overfitting due to its ensemble nature and the use of randomness. It typically performs well on a wide range of datasets without much hyperparameter tuning. With an eye towards scalability and technological advancement, this project not only aims to optimize crop production in the present but also anticipates future enhancements leveraging cutting-edge techniques and data sources to address the evolving challenges faced by the agricultural sector.

1.2 Aim of the Project

The aim of this project is to develop a ML model using the RF algorithm to predict crop yield based on diverse agricultural data. By leveraging information such as crop variety, soil characteristics, weather patterns, and agricultural practices, the model seeks to accurately forecast yield outcomes. Ultimately, this model aims to assist farmers, agricultural researchers, and policymakers in making informed decisions to optimize crop production and ensure food security.

1.3 Project Domain

ML is a branch of Artificial Intelligence (AI) where computers learn from data patterns and make predictions or decisions without being explicitly programmed. It involves algorithms that analyze data, identify patterns, and learn from them to make predictions or decisions. This process typically involves training a model on a dataset, fine-tuning it to improve performance, and then using it to make predictions or decisions on new data. ML has applications across various fields, from image recognition and natural language processing to recommendation systems and predictive analytics.

The project focuses on utilizing ML, specifically the RF algorithm, to predict crop yield. Falling within the domain of agricultural technology and data science, it leverages historical data comprising soil nutrient levels nitrogen, phosphorus, potassium, and pH and environmental variables like temperature and rainfall. The goal is to accurately forecast crop productivity, aligning with precision agriculture principles, where advanced analytics are used to optimize farming practices and resource allocation to achieve enhanced yield outcomes. Ultimately, the project aims to empower stakeholders in the agricultural sector with actionable insights to enhance efficiency and productivity in farming operations.

Within the agricultural landscape, precise yield forecasting holds significant importance for farmers, researchers. By harnessing ML techniques like RF, the project aims to uncover complex relationships between agronomic factors and crop yield variations. Through extensive dataset analysis, the model endeavors to provide actionable insights into optimal crop management strategies, facilitating informed decision-making processes. Ultimately, this project contributes to the ongoing evolution of agriculture, empowering stakeholders with the tools necessary to adapt to changing environmental conditions and improve overall agricultural productivity. Acknowledging the regional variability in agricultural practices and environmental conditions, the project will consider methods to adapt the predictive model to different geographical locations and crop types.

1.4 Scope of the Project

The project scope encompasses the development and implementation of a robust ML model to predict crop yield accurately. It involves gathering and preprocessing diverse datasets containing soil nutrient levels, environmental variables, and historical yield records. The scope extends to the selection and tuning of appropriate ML algorithms, with a focus on the RF technique for its ability to handle complex relationships in agricultural data. Additionally, the project involves rigorous testing and validation procedures to assess the model's performance and ensure its reliability in real-world scenarios.

Furthermore, the scope encompasses the integration of the predictive model into platforms to provide farmers and stakeholders with accessible and actionable insights. This involves designing user-friendly interfaces and APIs for seamless integration with decision support tools or farm management software. The project also includes considerations for scalability and adaptability, allowing for future enhancements and updates as new data becomes available or as agricultural practices evolve. Overall, the scope of the project aims to address the critical need for accurate crop yield prediction in agriculture while providing practical solutions that empower farmers to make informed decisions and optimize their farming operations efficiently.

Chapter 2

LITERATURE REVIEW

[1] M. Ashfaq et al, in 2024 explored the use of machine learning models and climate-NDVI data fusion for accurate wheat yield prediction. By analyzing historical yield data and considering various factors like climate, soil, and socioeconomic conditions, the study aims to enhance the precision of crop yield estimation, particularly focusing on wheat production in Pakistan. The research showcases the significance of yield detrending and aims to streamline the conventional manual estimation process. Through this approach, the study seeks to provide a reliable method for predicting wheat yield before the harvesting season, ultimately contributing to improved agricultural practices and food security.

[2] M. J. Hoque et al, in 2024 presented a novel approach to predicting crop yields by incorporating meteorological data and pesticide information using ML techniques. The study focuses on key crops in India and achieves high accuracy in yield forecasting through models like Gradient Boosting (GD). By analyzing data from reputable sources and implementing advanced methodologies like hyperparameter tuning, the research aims to enhance agricultural productivity, mitigate climate change risks, and improve food security. The findings demonstrate the potential of data-driven methods in sustainable agriculture, paving the way for more secure food availability and resilience to environmental challenges.

[3] A. Reyana et al, in 2023 presented"Accelerating Crop Yield: Multisensor Data Fusion and ML for Agriculture Text Classification" focuses on utilizing multisensor data fusion and ML techniques to enhance crop production in agriculture. It discusses the application of algorithms like RF for classifying agriculture text, aiming to improve accuracy in crop yield prediction and variety recommendations. The study addresses challenges in agriculture due to changing environmental conditions and poor practices, offering solutions to assist farmers in decision-making for increased productivity. Overall, the research emphasizes the importance of integrating sensor data and advanced technologies to optimize agricultural processes and boost

crop yields.

[4] P. Sharma et al, in 2023 focused on predicting crop yields using ML and deep learning techniques to address the challenge of increasing food demand due to population growth. By analyzing variables such as rainfall, crop type, meteorological conditions, area, production, and yield, the study aims to assist farmers in making informed decisions about crop selection and management during the growing season. The research evaluates various algorithms, including decision tree, random forest, XGBoost regression, convolutional neural network, and long-short term memory network, to achieve high accuracy in yield prediction. The findings aim to enhance agricultural productivity and sustainability for long-term food security.

[5] U. Shafi et al, in 2023 explored the use of remote sensing, ML, and UAV technology to predict crop yields and address food insecurity. The research team, led by experts from Pakistan and the USA, developed a regression model using XGBoost to predict grain yield based on vegetation indices. By leveraging innovative technologies like UAV monitoring and multi-spectral sensors, the study aims to provide accurate and timely predictions of crop yields on large farmlands. The findings highlight the potential of ML tools in enhancing food security efforts by enabling better crop yield forecasting and management strategies.

[6] H. R. Seireg et al, in 2022 explored the application of ensemble ML techniques using computer simulation data for predicting wild blueberry yields. By analyzing factors such as weather conditions, bee species densities, and feature selection, the study aims to enhance the accuracy of crop yield predictions. The research leverages a dataset generated by a simulation model and meteorological data from Maine, USA, and the Canadian Maritimes. Through the use of stacking and cascading techniques with various ML algorithms, the study demonstrates improved prediction accuracy compared to previous research. The findings contribute to advancing precision agriculture practices by offering insights into optimizing crop yield forecasting and management strategies.

[7] S. P. Raja et al, in 2022 focused on utilizing ML techniques for crop prediction based on environmental conditions. It discusses the use of feature selection and classification methods on a real-time dataset containing potato tuber yield, dry

matter yield, and starch yield. Various sampling techniques are applied during pre-processing to enhance prediction performance. The paper outlines the methodology, feature selection techniques, classification methods, and experimental design. By analyzing agricultural data and environmental factors, the study aims to provide valuable insights for farmers to make informed decisions about crop cultivation, ultimately improving agricultural productivity and efficiency.

[8] A. Sharma et al, in 2021 provided a thorough examination of the applications of ML in Precision Agriculture, focusing on the integration of AI and Internet of Things technologies. It discusses the impact of AI and IoT in agriculture, elucidates ML algorithms, reviews various ML applications in precision farming, evaluates IoT applications, and outlines challenges and future trends in AI for precision agriculture. Through case studies and analysis, the paper highlights the role of ML in optimizing crop production, resource management, and decision-making processes in the agricultural sector, ultimately contributing to sustainable and efficient farming practices.

[9] M. Rashid et al, in 2021 provided a detailed examination of ML applications in predicting crop yield, with a particular focus on palm oil production. By employing PRISMA guidelines and searching major databases, 223 relevant articles were selected for analysis. The review covers various aspects such as publication year, dataset information, features, prediction algorithms, and system performance. It offers insights into the current state of palm oil production, fundamentals of crop yield prediction, critical evaluation of ML algorithms, and the associated benefits and challenges. This study aims to pave the way for future research in improving crop yield prediction models, especially in the palm oil industry.

[10] Y. Alebele et al, in 2021 presented a novel approach for estimating rice grain yield using a combination of optical and SAR imagery, specifically Sentinel-2 vegetation indices and Sentinel-1 interferometric coherence data. The study employs Gaussian kernel regression to predict crop yield, demonstrating superior performance compared to other methods. By integrating these two types of data, the model achieves accurate and reliable predictions of rice yield in Xinghua county, China. The research contributes valuable insights into the potential of remote sensing technologies for enhancing agricultural productivity and monitoring crop

yields effectively.

The research papers collectively underscore the transformative potential of ML, remote sensing, and advanced data analytics in revolutionizing crop yield prediction and agricultural management practices. Leveraging methodologies ranging from regression models to deep learning algorithms and integrating diverse datasets encompassing environmental variables, soil characteristics, and satellite imagery, these studies offer innovative solutions to enhance prediction accuracy and optimize resource allocation. By providing actionable insights and recommendations, such as crop selection strategies and precision farming techniques, the research contributes to advancing agricultural productivity, mitigating risks associated with climate variability, and ensuring food security in the face of evolving environmental challenges.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system for crop yield prediction typically relies on traditional statistical methods and simplistic regression models. These approaches often lack the capacity to capture the complex interplay between various agronomic factors and environmental variables that influence crop productivity. Moreover, the reliance on manual data analysis and simplistic modeling techniques limits the accuracy and scalability of yield forecasting efforts.

In many agricultural settings, farmers make decisions based on their intuition, past experience, and limited access to data-driven insights. This reliance on subjective judgment can lead to suboptimal outcomes and inefficiencies in resource allocation. Furthermore, the lack of real-time data integration and predictive analytics tools hinders the ability to proactively address challenges such as pest outbreaks, soil degradation, and adverse weather conditions. The existing system for crop yield prediction is characterized by its reliance on traditional methods and its limited capacity to harness the potential of advanced data analytics and ML techniques.

The disadvantages of the existing system include its vulnerability to inaccuracies due to oversimplified models and the absence of dynamic data integration. These limitations impede the system's ability to provide timely and precise predictions, hindering farmers' capacity to mitigate risks and optimize crop management strategies effectively. Thus, there is a pressing need to transition towards more sophisticated predictive modeling approaches that leverage big data and AI to address these shortcomings and empower farmers with actionable insights for sustainable agricultural practices.

3.2 Proposed System

The proposed system presents a holistic approach to accurate crop yield prediction, anchored by the deployment of a ML model based on the RF algorithm. Trained on extensive datasets encompassing vital agronomic factors such as soil nutrient levels (nitrogen, phosphorus, potassium, Potential of Hydrogen (PH)) and environmental variables (temperature, rainfall), this model stands as a beacon of predictive precision within the agricultural domain. Its implementation marks a departure from traditional methods, offering farmers and stakeholders a robust tool for forecasting crop productivity with unprecedented accuracy.

Complementing the ML backbone is a user-friendly interface, strategically crafted to streamline interaction and facilitate intuitive utilization of predictive capabilities. Through this interface, users can seamlessly input relevant data and receive instantaneous yield predictions tailored to specific scenarios or conditions. This accessibility ensures that the transformative potential of advanced analytics and predictive modeling is readily accessible to farmers, agronomists, and decision-makers, empowering them to make informed choices and optimize farming practices effectively.

Beyond its predictive prowess and user-centric design, the proposed system boasts advantages that resonate deeply within the agricultural landscape. By harnessing ML and real-time data integration, it fosters proactive decision-making and resource optimization, driving tangible improvements in agricultural efficiency and productivity. Furthermore, its scalability and adaptability promise continuous refinement and enhancement, positioning the system at the vanguard of innovation in crop yield prediction. In sum, the proposed system represents a quantum leap in agricultural technology, offering holistic solutions to the multifaceted challenges confronting modern farming practices.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of the proposed system lies in its potential to yield significant returns on investment for stakeholders in the agricultural sector. By accurately predicting crop yield, the system enables farmers to optimize resource allocation, minimize input costs, and maximize profits. Through informed decision-making facilitated by the predictive model, farmers can mitigate risks associated with crop failure and market fluctuations, thus enhancing overall economic resilience.

Moreover, the implementation of the proposed system offers long-term cost savings by reducing reliance on traditional, resource-intensive farming practices. By leveraging ML and data-driven insights, farmers can optimize fertilizer and water usage, leading to more efficient use of resources and lower operational expenses. Additionally, the system's scalability and adaptability ensure that it can evolve alongside changing agricultural landscapes, providing sustained economic benefits over time. Overall, the economic feasibility of the proposed system lies in its potential to drive profitability and sustainability in agricultural operations, ultimately contributing to the economic well-being of farmers and stakeholders in the industry.

3.3.2 Technical Feasibility

The technical feasibility of building a ML model using the random forest algorithm to predict crop yield based on historical crop data, weather data, and other relevant factors is quite high. Random forest is a robust algorithm capable of handling complex datasets with both numerical and categorical variables, making it suitable for agricultural prediction tasks. Moreover, the availability of historical crop data and weather records, often collected by agricultural agencies or research institutions, facilitates model training and validation. However, challenges may arise in data preprocessing, such as handling missing values, dealing with outliers, and ensuring data quality and consistency. Additionally, integrating multiple data sources, such as crop yield records, weather data from various sources, and soil information, may require careful data integration and cleaning. Furthermore, model performance optimization through hyperparameter tuning and feature selection can be computationally intensive but feasible with modern computing resources. Overall, while there are technical challenges to overcome, the project's feasibility is high given the availability of relevant data and the suitability of RF for agricultural prediction tasks.

3.3.3 Social Feasibility

The social feasibility of implementing a ML model to predict crop yield using historical crop data, weather information, and other relevant factors is significant, but it also presents certain considerations. On one hand, such a system could offer valuable insights to farmers, agricultural policymakers, and stakeholders in optimizing crop production, mitigating risks, and improving food security. By providing accurate predictions of crop yields, farmers can make informed decisions regarding planting strategies, resource allocation, and crop management practices, potentially increasing productivity and profitability. Additionally, policymakers can utilize these predictions to formulate evidence-based agricultural policies and allocate resources more effectively to support farmers and ensure food sufficiency.

However, there are social factors to consider, including accessibility and adoption of the technology by different stakeholders. While ML models can offer powerful tools for decision-making, their effectiveness relies on factors such as data accessibility, technological literacy, and trust in the system. Ensuring equitable access to the technology and providing training and support to users, especially farmers in rural or resource-constrained areas, is crucial for widespread adoption and benefit realization. Furthermore, addressing concerns related to data privacy, ownership, and transparency is essential to building trust and acceptance among stakeholders. Overall, while the social benefits of the project are substantial, addressing these considerations will be vital for its successful implementation and impact.

3.4 System Specification

3.4.1 Hardware Specification

- **Central Processing Unit (CPU):**
  - Processor 12th Gen Intel(R) Core(TM) i5-1235U
- **Memory (RAM):**
  - DDR4 (Minimum 32GB)
- **Storage:**
  - Solid State Drive (SSD)

3.4.2 Software Specification

- **Programming Language:**
  - Python (version 3.6 or higher)
- **Integrated Development Environment (IDE):**
  - Visual Studio Code or Jupyter Notebooks

- **Cloud Platform :**
  – Google Colab

3.4.3 Standards and Policies

Google Colab

Google Colab, short for Colaboratory, is an online platform provided by Google that offers a cloud-based environment for running Python code, particularly popular for its integration with Jupyter Notebooks. Unlike local environments, Google Colab operates entirely in the cloud, enabling collaborative coding and resource-intensive tasks without the need for extensive hardware specifications. While Google Colab doesn't adhere to a specific ISO/IEC standard like other applications, its usage aligns with general cybersecurity and data protection principles. Users are encouraged to follow best practices in securing their notebooks, such as restricting access and managing data privacy according to their organizational or institutional standards.

Standard Used: ISO/IEC 27001

Visual Studio Code

Visual Studio Code (VS Code) is a source-code editor developed by Microsoft, known for its versatility and support for various programming languages. While it doesn't adhere to a specific ISO/IEC standard, Microsoft, the developer of Visual Studio Code, maintains a commitment to security and privacy in its software development practices. Therefore, while Visual Studio Code itself may not have a direct standard, users can benefit from Microsoft's broader adherence to industry-recognized standards for information security and data protection. It's advisable for users to implement their own security measures and adhere to relevant standards based on their specific use cases and organizational requirements.

Chapter 4

METHODOLOGY

4.1 Architecture of Crop Yield Prediction System

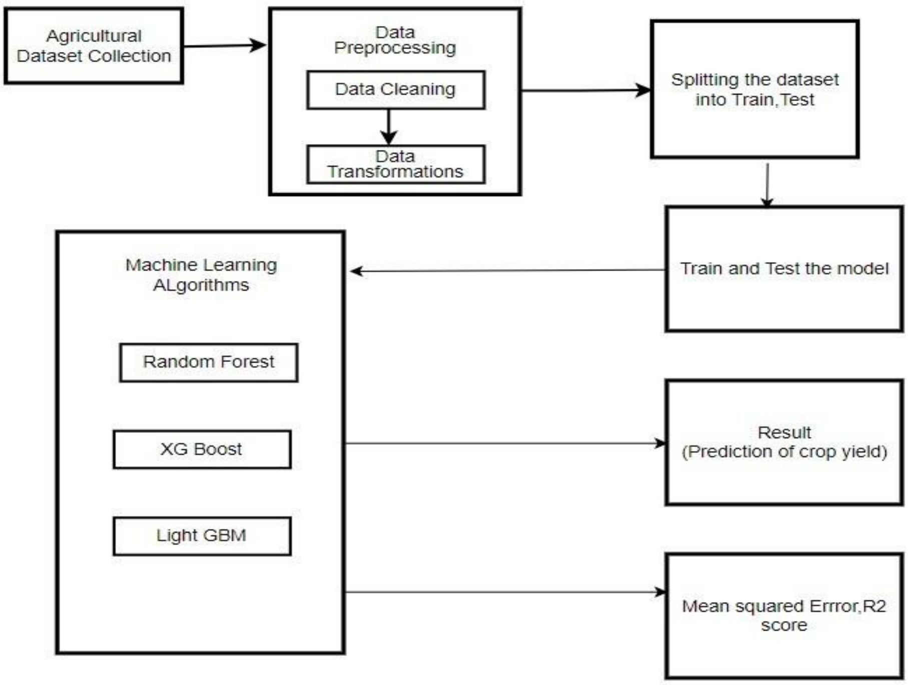

Figure 4.1: **Architecture Diagram of Crop Yield Prediction**

The figure 4.1 depicts the architecture diagram of the crop yield prediction system. It starts with the collection, preprocessing, and cleaning of agricultural dataset to ensure data readiness. The dataset is then split into train and test sets for training and testing ML algorithms such as RF, XGBoost and Light GBM. These algorithms are utilized to predict crop yield, with performance evaluated using metrics like Mean Squared Error(MSE), R- Squared score (R2 Score) and Root Mean Squared Error (RMSE).

4.2 Design Phase

4.2.1 Data Flow Diagram

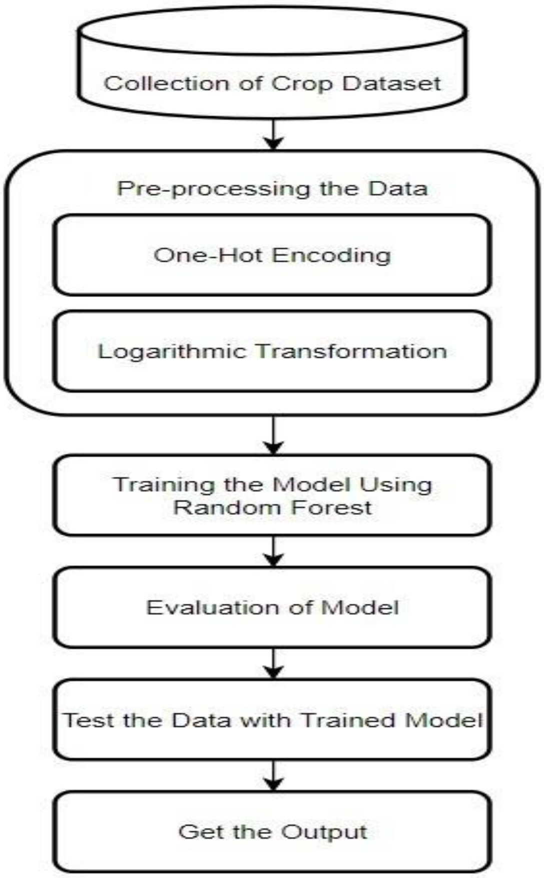

Figure 4.2: **Data Flow Diagram**

The figure 4.2 explains the flow of data in the crop yield prediction system. Firstly, a dataset with various features is collected and it is pre-processed to clean, handle missing values, remove outliers, and normalize it. Categorical variables are encoded using one-hot encoding, and a logarithmic transformation is applied to reduce skewness and stabilize variance. Then, a RF model is trained on the transformed data and tested. The model's performance is evaluated using metrics, and it's tested with new data. Finally, the model's predictions inform decision-making and optimize agricultural practices.

4.2.2 Use Case Diagram

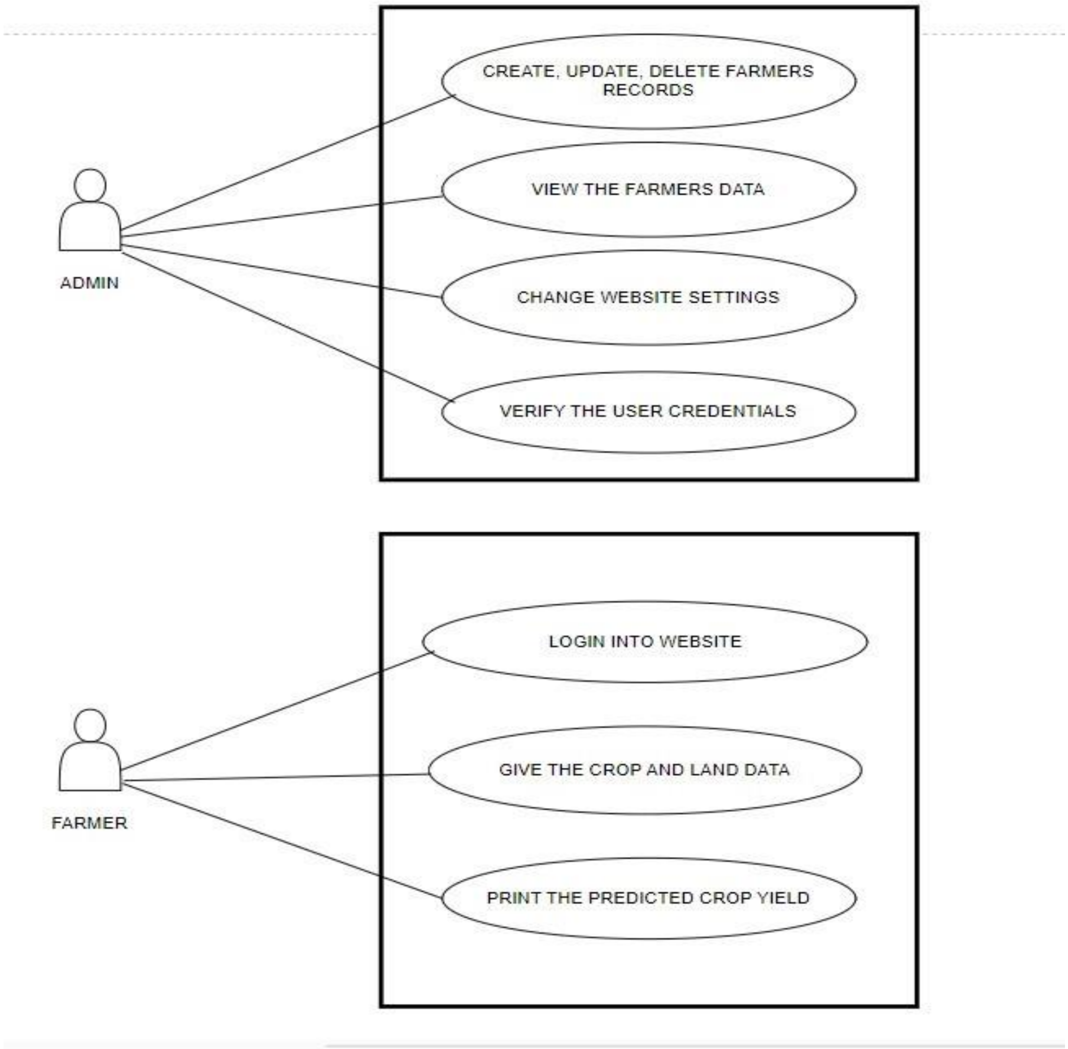

Figure 4.3: **Use Case Diagram**

The figure 4.3 depicts the usecase diagram of the crop yield prediction system. Administrators hold the responsibility for managing farmer records by creating, updating, and deleting them, as well as overseeing website settings and verifying user credentials. Conversely, farmers interact with the system by logging in, inputting crop and land data, and printing the predicted crop yield generated by the system. This diagram illustrates the interactions between actors and the system, delineating the administrative tasks and farmer actions, which collectively facilitate the functionality and operation of the crop yield prediction system.

4.2.3 Sequence Diagram

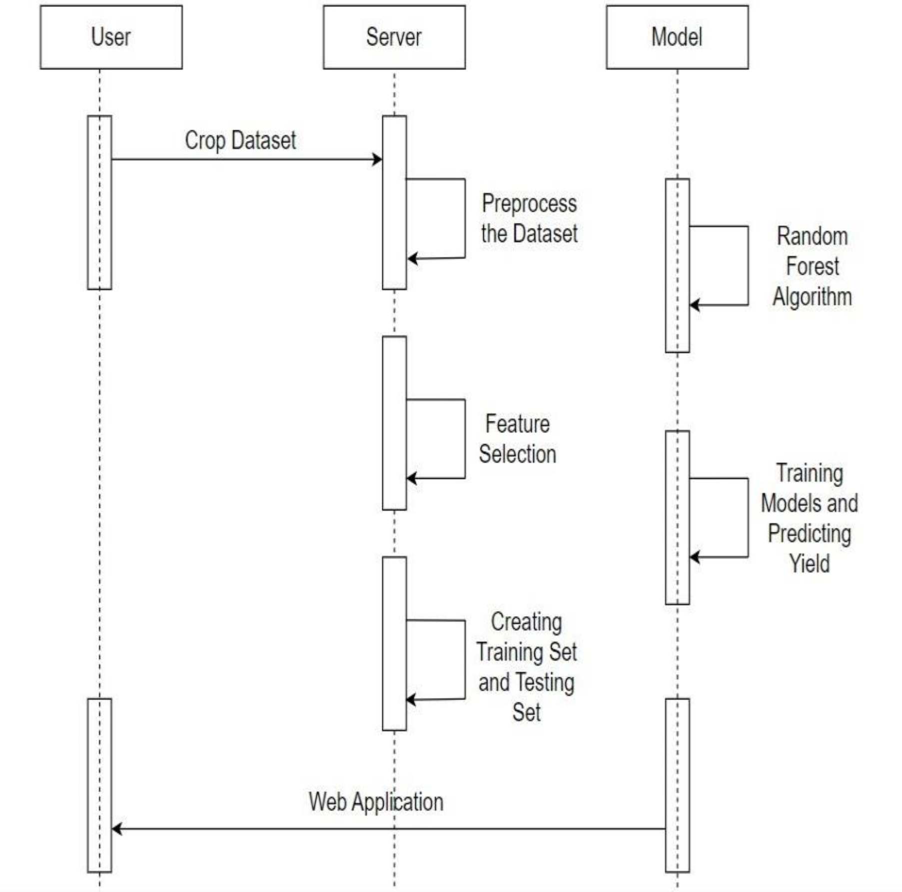

Figure 4.4: **Sequence Diagram**

The figure 4.4 depicts the sequence diagram for crop yield prediction system. The farmer representing the user, initiates the process by providing crop data, including soil type, temperature, and rainfall, to the crop yield prediction. The system object, symbolizing the server undertakes tasks such as pre-processing, encoding categorical variables, and transforming the data. Subsequently, the server communicates with the RF model, by sending the prepared data. The model generates a prediction based on the input and forwards it back to the server. Finally, the server displays the prediction to the user. This diagram illustrates the step-by-step interactions between the user, server, and model, offering clarity on the crop yield prediction process and aiding in the detection of potential issues.

4.2.4 Entity Relationship Diagram

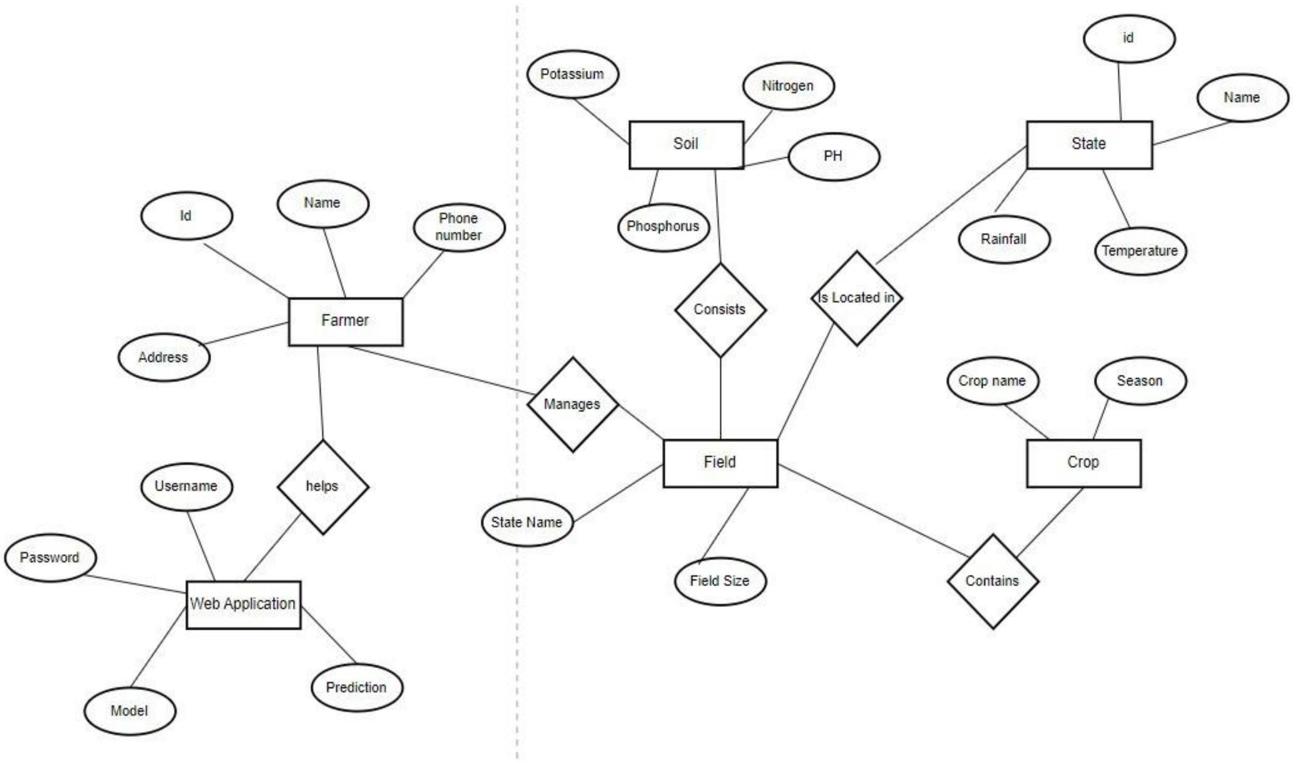

Figure 4.5: **Entity Relationship Diagram**

The figure 4.5 represents the entity relationship diagram for crop yield prediction system. It includes entities such as farmer, crop, web application, soil with attributes such as crop attributes and soil attributes etc. The relationships between the entities would represent the associations between the farmer, crop, and prediction on web application, with the farmer providing crop data for the prediction. The diagram would provide a visual representation of the entities, attributes, and relationships involved in the crop yield prediction system, helping to clarify the structure of the database and the relationships between the entities.

4.2.5 Activity Diagram

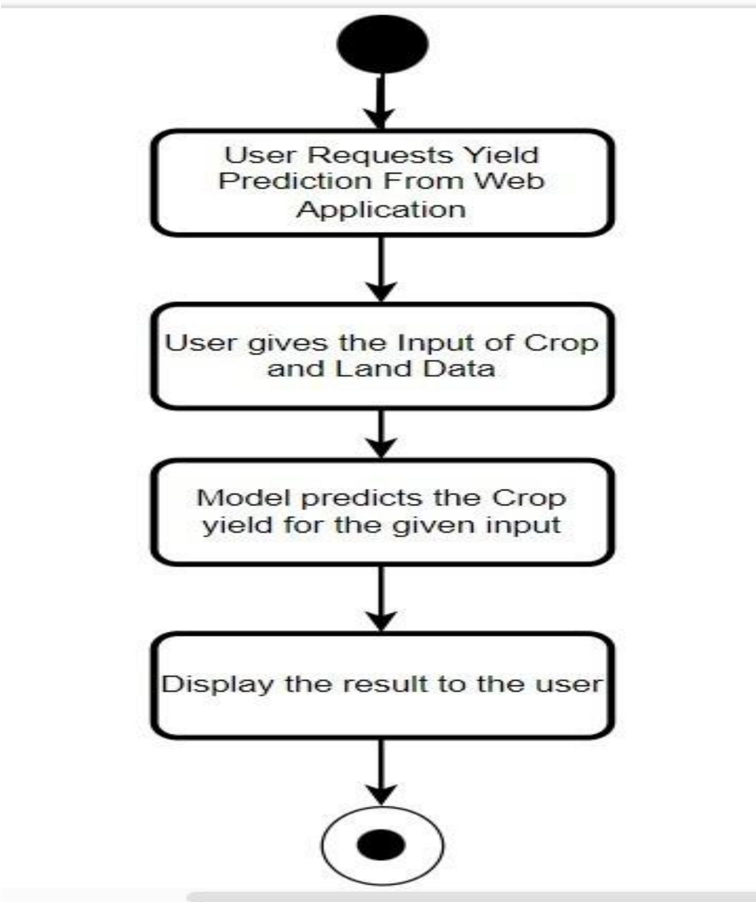

Figure 4.6: **Activity Diagram**

The figure 4.6 depicts the activity diagram for crop yield prediction system via a web application. The user initiates the process by requesting a yield prediction. Upon this request, the user provides input data regarding the crop and land. Subsequently, the model, tasked with predicting crop yield, processes the provided data and generates predictions based on established algorithms and trained models. Once the prediction is computed, the result is displayed to the user, completing the cycle. This diagram encapsulates the sequential flow of actions, from user initiation to result presentation, within the crop yield prediction system, facilitating comprehension of the process's workflow and user interaction points.

4.3 Algorithm & Pseudo Code

4.3.1 Enhanced Random Forest Algorithm

1. Preprocess the data by loading necessary libraries, inspecting the dataset, and performing exploratory data analysis through visualization.
2. Engineer features like NPK Ratio and Temperature-Rainfall Interaction to enhance the dataset.
3. Visualize the distribution of numerical attributes and the count of categorical attributes to understand data characteristics.
4. Encode categorical variables using one-hot encoding for compatibility with machine learning algorithms.
5. Split the dataset into training and testing sets to assess model performance.
6. Train a Random Forest regressor model using the training data.
7. Evaluate the trained model's performance by making predictions on the test set and calculating relevant metrics.
8. Prompt the user to input values for each feature, including categorical variables like Crop, State Name, and Crop Type.
9. Map the user inputs to the corresponding one-hot encoded columns in the dataset.
10. Use the trained model to predict crop yield based on the user's input.
11. Save the trained Random Forest model for future use in deployment.

4.3.2 Pseudo Code

```
# Step 1: Data Preprocessing
Import necessary libraries (pandas, numpy, etc.)
Load dataset into a DataFrame
Perform exploratory data analysis (EDA):
    - Inspect data types, missing values, and basic statistics
    - Visualize data distributions and correlations between variables
    - Engineer new features if necessary (e.g., NPK Ratio, Temperature-Rainfall Interaction)
    # Step 2: Data Encoding
Encode categorical variables using one-hot encoding:
    - Identify categorical columns
    - Convert categorical variables into numerical format suitable for machine learning
# Step 3: Data Splitting
Split dataset into features (X) and target variable (y)
Split data into training and testing sets:
    - Reserve a portion of the data for testing the trained models performance
    # Step 4: Model Training
```

```
Train a Random Forest regressor model:
   - Instantiate the Random Forest regressor with desired hyperparameters
    - Fit the model to the training data
    # Step 5: Model Evaluation
Evaluate the trained models performance:
    - Make predictions on the test set
    - Calculate evaluation metrics such as Mean Squared Error, R-squared, and Root Mean Squared
        Error
    - Assess the models ability to generalize to unseen data
# Step 6: User Input
Prompt the user to input values for each feature:
    - Request information such as N, P, K levels, pH, rainfall, temperature, Crop, State Name, and
        Crop_Type
    - Ensure proper data validation and handling of user inputs
# Step 7: Prediction
Utilize the trained model to predict crop yield based on user input:
    - Map user inputs to corresponding features in the dataset
    - Use the trained model to predict the crop yield
# Step 8: Save Model
Save the trained Random Forest model for future use:
    - Serialize the model using a suitable format (e.g., joblib, pickle)
    - Store the model file in
```

4.4 Module Description

4.4.1 Collection of Dataset

The dataset used in training the model comprises historical data essential for predicting crop yield accurately. It includes 99849 rows and 13 columns, encompassing a wide array of information spanning various factors crucial for understanding and forecasting crop productivity. This comprehensive dataset encapsulates not only soil nutrient levels like Nitrogen (N), Phosphorus (P), and Potassium (K), but also environmental variables such as rainfall and temperature. Additionally, categorical variables such as State Name, Crop Type, and Crop provide contextual information about geographical locations and specific crop types and this enables robust analysis and model training.

The columns encompass key factors directly impacting crop productivity. Nitrogen (N), Phosphorus (P), and Potassium (K) levels in the soil are pivotal indicators of soil fertility and plant nutrition, essential for optimizing fertilization strategies to ensure optimal crop growth and yield. Soil pH influences nutrient availability and plant uptake, significantly affecting overall crop health and productivity. Rainfall and temperature data provide insights into climatic conditions, aiding in the assessment of water availability and temperature stress, both crucial for crop development and yield. Additionally, variables such as Area in hectares enable consideration of cultivation scale, facilitating yield estimation per unit area. Through meticulous collection and analysis of these datasets, the aim is to develop robust predictive models to empower farmers with valuable insights for optimizing agricultural practices and maximizing crop yield.

Quality control measures were implemented to validate and standardize the collected data, ensuring consistency and reliability across all variables and observations. This comprehensive approach to dataset collection aimed to capture the complexity and variability inherent in agricultural systems, laying the foundation for robust analysis and model development to support informed decision-making and sustainable agricultural practices.

The dataset utilized for the project is collected from the Kaggle website: https://www.kaggle.com/datasets/sriharikatare/indian-crop-production.

```
First few rows of the dataset:
```

	Unnamed: 0	State_Name	Crop_Type	Crop	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	Yield_ton_per_hec
0	0	andhra pradesh	kharif	cotton	120	40	20	5.46	654.34	29.266667	7300.0	9400.0	1.287671
1	1	andhra pradesh	kharif	horsegram	20	60	20	6.18	654.34	29.266667	3300.0	1000.0	0.303030
2	2	andhra pradesh	kharif	jowar	80	40	40	5.42	654.34	29.266667	10100.0	10200.0	1.009901
3	3	andhra pradesh	kharif	maize	80	40	20	5.62	654.34	29.266667	2800.0	4900.0	1.750000
4	4	andhra pradesh	kharif	moong	20	40	20	5.68	654.34	29.266667	1300.0	500.0	0.384615

Figure 4.7: **Head of Dataset**

Continous Attributes	N, P, K, pH, Rainfall, Temperature, Area_in_hectares, Production_in_tons, Yield_per_hec
Categorical Attributes	State_Name, Crop_type, Crop

Table 4.1: **Various Attributes of Dataset**

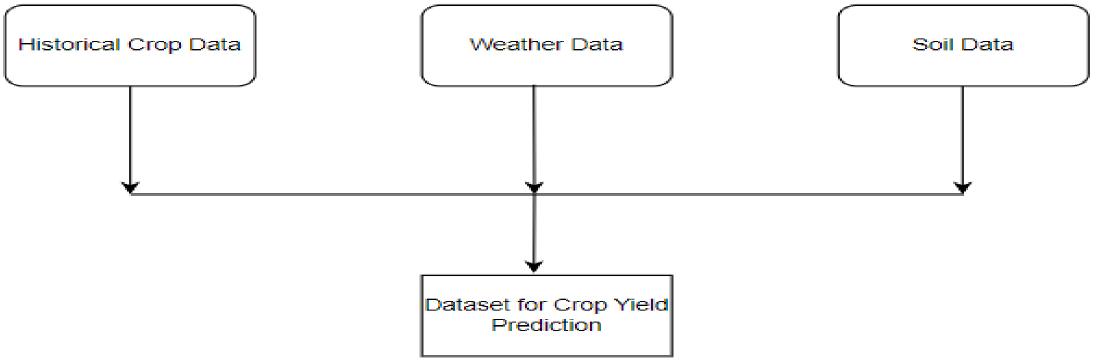

Figure 4.8: **Dataset Collection**

4.4.2 Pre-Processing the Dataset

Preprocessing is a crucial step in preparing data for analysis and modeling. It involves transforming raw data into a format suitable for ML algorithms, improving model performance and accuracy. In the context of the dataset used for predicting crop yield, preprocessing entails handling missing values, scaling numerical features, and encoding categorical variables. For the dataset, one-hot encoding is applied to categorical variables, while logarithmic transformation is used for numerical attributes.

One-Hot Encoding: One-hot encoding is a technique applied to categorical variables in which each category is represented by a binary dummy variable. This process is particularly useful for categorical attributes such as Crop Type, State Name, and Crop in crop yield prediction. By converting categorical variables into numerical format, one-hot encoding allows the model to capture relationships between different crop types and regions without imposing any ordinality or hierarchy. This enables the model to effectively incorporate categorical variables into the predictive process, enhancing its ability to generalize and make accurate predictions.

Logarithmic Transformation: Logarithmic transformation is a mathematical operation applied to numerical data to normalize skewed distributions and reduce the influence of extreme values. In crop yield prediction, it involves taking the logarithm of numerical attributes such as Nitrogen, Phosphorus, Potassium, rainfall, and temperature. By transforming these attributes, logarithmic transformation helps to mitigate skewness. It makes the data more symmetrically distributed and improves the model's ability to capture relationships between variables. This preprocessing step enhances predictive accuracy and facilitates the identification of meaningful patterns in the data.

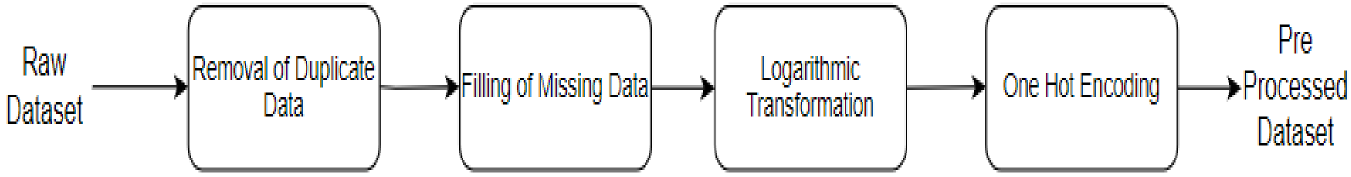

Figure 4.9: **Pre-Processing the Dataset**

4.4.3 Training the Model

For training the dataset, Random Forest algorithm under ML was employed to predict crop yield. RF is a powerful ensemble learning method widely used for regression and classification tasks. It constructs multiple decision trees during training and outputs the average prediction of the individual trees for regression tasks. Where each tree in the forest is trained independently on a random subset of the dataset and makes its own prediction. The final prediction of the RF is determined by averaging or voting the predictions of all the individual trees. In the context of predicting crop yield, RF offers several advantages:

Ensemble Learning: RF builds multiple decision trees, each trained on a random subset of the dataset. By combining the predictions of these individual trees, RF reduces the risk of overfitting and improves generalization performance.

Robustness to Noise: RF is inherently robust to noise and outliers in the data. Since it aggregates predictions from multiple trees, it can handle noisy data effectively, making it suitable for real-world agricultural datasets that may contain noise or outliers.

Feature Importance: RF provides a measure of feature importance, indicating which variables have the most significant impact on crop yield prediction. This insight can help identify key factors influencing crop productivity, guiding farmers in decision-making and resource allocation.

Non-linear Relationships: RF can capture complex non-linear relationships between input variables and crop yield. It is capable of modeling interactions and dependencies among various factors such as soil nutrients, climate conditions, and crop management practices, leading to more accurate predictions.

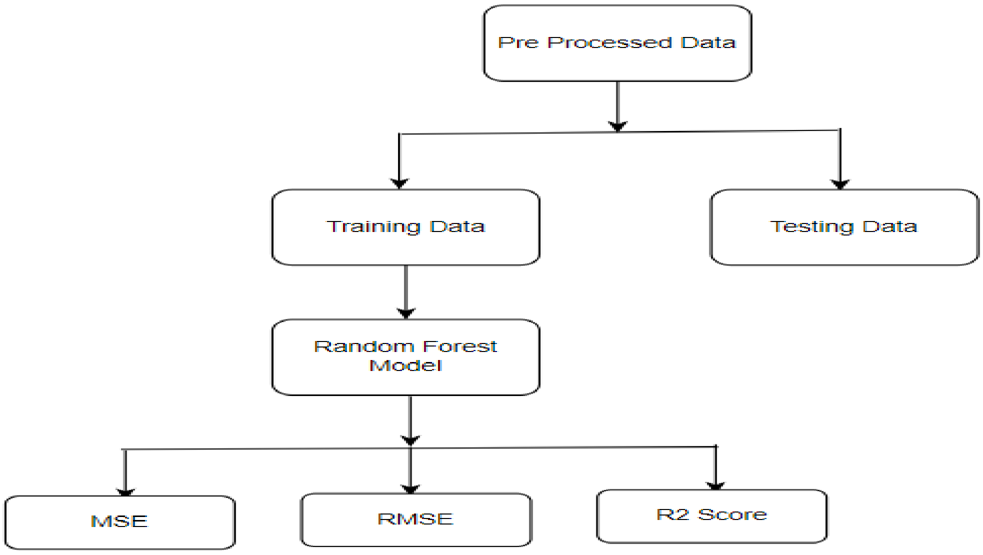

Figure 4.10: **Training of Model**

4.4.4 Testing the Model

For testing the crop yield prediction model, a rigorous evaluation process is employed to assess its proficiency in accurately forecasting crop yields based on the provided features. The model undergoes testing using a separate and previously unseen test dataset, distinct from the training data, to evaluate its generalization capabilities. During this phase, each sample from the test dataset is processed through the model's learned representations, generating predictions based on the knowledge acquired during training. The model's predictions are then compared against the actual crop yields from the test dataset to determine its accuracy and effectiveness in predicting crop yields across different scenarios and conditions.

The output of the testing phase encompasses various evaluation metrics tailored to crop yield prediction, including MSE, RMSE, and R2 Score. These metrics offer insights into the model's performance in accurately estimating crop yields and quantifying the proportion of variance in the crop yield that is predictable from the independent variables. By thoroughly evaluating the model's performance using these metrics, a comprehensive assessment of its predictive capabilities is obtained, enabling the identification of strengths and areas for potential improvement. This ensures the model's reliability in predicting crop yields and provides valuable insights for optimizing agricultural practices and maximizing crop productivity.

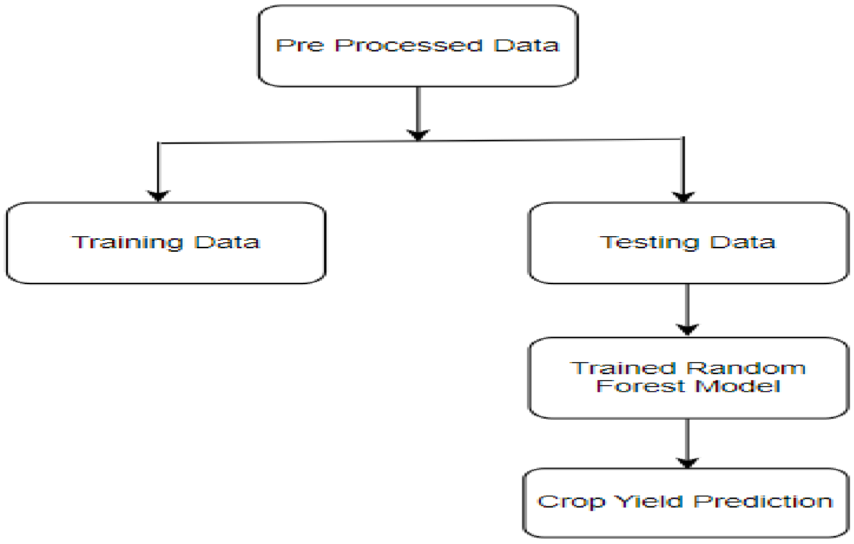

Figure 4.11: **Testing the Model**

4.5 Steps to execute/run/implement the project

4.5.1 Setup Environment

- Ensure Python is installed.
- Install required packages: NumPy, Pandas, scikit-learn, TensorFlow.
- Set up a working directory for the project.

4.5.2 Download Dataset

- Obtain the dataset containing historical crop yield records, soil nutrient levels, and environmental variables.
- Place the dataset in a folder accessible to the project.

4.5.3 Project Structure

- Create a structured project folder.
- Include the python script and a subfolder for the dataset.

4.5.4 Data Preprocessing

- Preprocess the dataset to handle missing values and normalize numerical features.
- Perform feature engineering to extract relevant information for crop yield prediction.

4.5.5 Model Development

- Develop a Random Forest Regression model using scikit-learn.
- Define model hyperparameters such as number of trees, maximum depth, and minimum samples split.

4.5.6 Model Training

- Train the Random Forest Regression model using the preprocessed dataset.
- Specify attributes used in the model training which are passed as parameters.

4.5.7 Model Evaluation

- Evaluate the trained model's performance on a validation set using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RSME) and R-squared Score.

4.5.8 Results and Analysis

- Display or log training and validation results, including evaluation metrics and model performance.
- Analyze the model's ability to predict crop yields accurately and identify areas for further improvement.

4.5.9 Model Deployment

- Save the trained model to a file for future use and deployment.
- Implement the model into an interface for users to input relevant data and receive crop yield predictions.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

	Unnamed: 0	State_Name	Crop_Type	Crop	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	Yield_ton_per_hec
0	0	andhra pradesh	kharif	cotton	120	40	20	5.46	654.34	29.266667	7300.0	9400.0	1.287671
1	1	andhra pradesh	kharif	horsegram	20	60	20	6.18	654.34	29.266667	3300.0	1000.0	0.303030
2	2	andhra pradesh	kharif	jowar	80	40	40	5.42	654.34	29.266667	10100.0	10200.0	1.009901
3	3	andhra pradesh	kharif	maize	80	40	20	5.62	654.34	29.266667	2800.0	4900.0	1.750000
4	4	andhra pradesh	kharif	moong	20	40	20	5.68	654.34	29.266667	1300.0	500.0	0.384615
...	...	...	...	...	...	...	...	...	...	...	...	...	...
99844	99844	west bengal	rabi	wheat	60	30	30	6.70	152.54	22.280000	2013.0	5152.0	2.559364
99845	99845	west bengal	summer	maize	80	40	20	5.68	182.50	29.200000	258.0	391.0	1.515504
99846	99846	west bengal	summer	rice	80	40	40	5.64	182.50	29.200000	105.0	281.0	2.676190
99847	99847	west bengal	rabi	rice	80	40	40	5.42	152.54	22.280000	152676.0	261435.0	1.712352
99848	99848	west bengal	rabi	sesamum	30	15	30	6.54	152.54	22.280000	244.0	95.0	0.389344

99849 rows × 13 columns

Figure 5.1: **Historial Crop Dataset**

The figure 5.1 illustrates the input of the system, which comprises a diverse dataset containing soil nutrient levels, environmental variables, and historical crop yield records. This dataset encompasses various agronomic factors such as nitrogen, phosphorus, potassium, pH levels, temperature, rainfall, and past yield outcomes, providing comprehensive information for accurate crop yield prediction.

5.1.2 Output Design

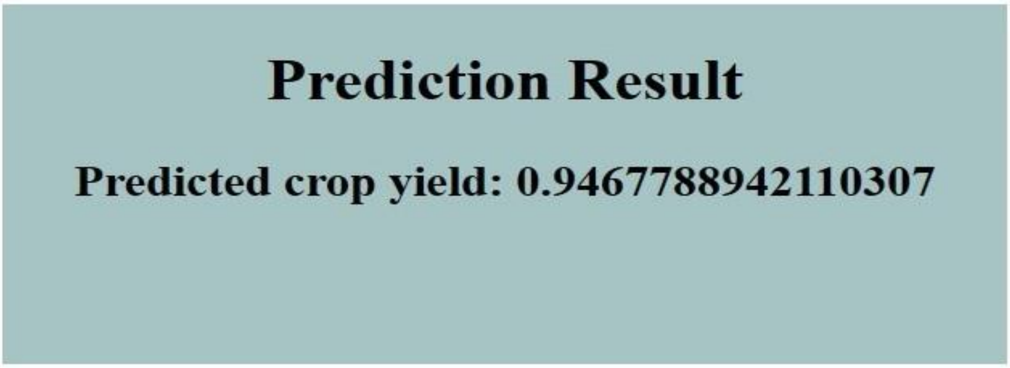

Figure 5.2: **Prediction of Crop Yield**

The figure 5.2 displays a predictive model evaluation for crop yield estimation. By analyzing key agronomic features and environmental variables, the model determines the predicted crop yield, aiding in effective decision-making for agricultural management.

5.2 Testing

The program undergoes rigorous testing to ensure its accuracy, reliability, and functionality. Various testing methodologies, including unit testing, assess individual components, while integration testing evaluates interactions between these components. Specialized testing for preprocessing techniques and ML models validates their effectiveness. Additionally, robustness testing examines the program's resilience to adverse conditions. Continuous testing and validation are integral to ensuring the program's efficacy in predicting crop yield accurately from diverse datasets, while maintaining high standards of precision and performance.

5.3 Types of Testing

5.3.1 Unit Testing

Unit testing is a software testing approach that involves the evaluation of individual units or components within a software application in isolation. These units typically refer to the smallest testable parts of the code, such as functions, methods, or procedures. The primary objective of unit testing is to confirm that each unit behaves as intended, independently of the rest of the application.

Input

```
# Apply logarithmic transformation
log_transformed_data = data.copy()
log_transformed_data[numerical_attributes] = log_transformed_data[numerical_attributes]
.apply(lambda x: np.log1p(x))
log_transformed_data.head()
```

Test result

	Unnamed: 0	State_Name	Crop_Type	Crop	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	Yield_ton_per_hec	N
0	0	andhra pradesh	kharif	cotton	4.795791	3.713572	3.044522	1.865629	6.485154	3.410047	7300.0	9.148571	1.287671	
1	1	andhra pradesh	kharif	horsegram	3.044522	4.110874	3.044522	1.971299	6.485154	3.410047	3300.0	6.908755	0.303030	
2	2	andhra pradesh	kharif	jowar	4.394449	3.713572	3.713572	1.859418	6.485154	3.410047	10100.0	9.230241	1.009901	
3	3	andhra pradesh	kharif	maize	4.394449	3.713572	3.044522	1.890095	6.485154	3.410047	2800.0	8.497195	1.750000	
4	4	andhra pradesh	kharif	moong	3.044522	3.713572	3.044522	1.899118	6.485154	3.410047	1300.0	6.216606	0.384615	

Figure 5.3: **Result of Unit Test**

5.3.2 Integration Testing

Integration testing is a software testing methodology that focuses on evaluating the interactions and dependencies between different components within a larger system. The goal of integration testing is to ensure that individual components, when combined, function correctly as a cohesive unit. This type of testing helps uncover errors related to data flow, control flow, and communication between modules.

Input

```
# Feature Creation
# NPK Ratio
data['NPK_Ratio'] = data['N'] / (data['P'] + data['K'])

# Temperature-Rainfall Interaction
data['Temp_Rainfall_Interaction'] = data['temperature'] * data['rainfall']

# Display the updated dataset with derived features
data.head()
```

Test result

	Unnamed: 0	State_Name	Crop_Type	Crop	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	Yield_ton_per_hec	NPK_Ratio	Temp_Rain
0	0	andhra pradesh	kharif	cotton	120	40	20	5.46	654.34	29.266667	7300.0	9400.0	1.287671	2.000000	
1	1	andhra pradesh	kharif	horsegram	20	60	20	6.18	654.34	29.266667	3300.0	1000.0	0.303030	0.250000	
2	2	andhra pradesh	kharif	jowar	80	40	40	5.42	654.34	29.266667	10100.0	10200.0	1.009901	1.000000	
3	3	andhra pradesh	kharif	maize	80	40	20	5.62	654.34	29.266667	2800.0	4900.0	1.750000	1.333333	
4	4	andhra pradesh	kharif	moong	20	40	20	5.68	654.34	29.266667	1300.0	500.0	0.384615	0.333333	

Figure 5.4: **Result of Integration Test**

5.3.3 Functional Testing

Functional testing is a crucial phase in software testing that focuses on verifying whether a software application or system performs its intended functions as specified in the requirements. It aims to ensure that the software meets the functional expectations and behaves according to the design specifications. During functional testing, the application is tested against its functional requirements, and test cases are designed to cover various scenarios, inputs, and system states.

Input

```
# Split data into features and target variable
X = encoded_data.drop(['Production_in_tons'], axis=1)  # Features
y = encoded_data['Production_in_tons']  #Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions and evaluate
rf_y_pred = rf_model.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_y_pred)
rf_r2 = r2_score(y_test, rf_y_pred)
rf_rmse = np.sqrt(rf_mse)

print("Random Forest:")
print("Mean Squared Error:", rf_mse)
print("R-squared:", rf_r2)
print("Root Mean Squared Error:", rf_rmse)
```

Test result

```
Random Forest:
Mean Squared Error: 47674917.95146866
R-squared: 0.9967375433339981
Root Mean Squared Error: 6904.702596887766
```

Figure 5.5: **Result of Functional Test**

5.3.4 Test Result

```
# Set the mapped columns to 1
new_data[crop_column] = 1 if crop_column in feature_names else 0
new_data[crop_type_column] = 1 if crop_type_column in feature_names else 0
new_data[state_name_column] = 1 if state_name_column in feature_names else 0

# Set all other one-hot encoded columns to 0
for column in feature_names:
    if column.startswith(('Crop_', 'Crop_Type_', 'State_Name_')):
        if column not in [crop_column, crop_type_column, state_name_column]:
            new_data[column] = 0

# Convert the dictionary into a DataFrame with a single row
new_data_df = pd.DataFrame([new_data])

# Reorder DataFrame columns based on feature names
new_data_df = new_data_df[feature_names]

# Make predictions on new data
new_predictions = rf_model1.predict(new_data_df)

# Print the predicted crop yield
print("Predicted crop yield:", new_predictions[0])
```

```
Please enter the values for each feature:
N: 80
P: 40
K: 20
pH: 5.62
rainfall: 654.34
temperature: 29.266667
Area_in_hectares: 2800.0
NPK_Ratio: 1.3333333
Temp_Rainfall_Interaction: 19150.350667
Crop: maize
State_Name: andhra pradesh
Crop_Type: kharif
Predicted crop yield: 2.755817266498499
```

Figure 5.6: **Result Obtained by Testing**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system, leveraging a RF algorithm to predict crop yield based on historical crop data, weather information, and other relevant factors, offers several efficiency benefits. Firstly, by automating the prediction process, the system enables farmers and agricultural stakeholders to make data-driven decisions swiftly and accurately. This automation reduces the time and effort required for manual analysis, allowing farmers to focus more on other critical tasks. Moreover, the system's ability to handle large and diverse datasets efficiently enhances its scalability, enabling it to process extensive historical data and incorporate various factors affecting crop yield.

Additionally, the use of ML techniques like RF ensures robustness and adaptability to different agricultural contexts and regions. By providing timely and accurate predictions, the system empowers farmers to optimize resource allocation, plan crop rotations, and mitigate risks effectively, leading to improved productivity and profitability. Furthermore, the system's user-friendly interface for inputting data and obtaining predictions enhances its accessibility and usability for farmers of varying technical backgrounds. Overall, the proposed system offers efficiency gains in decision-making, resource management, and risk mitigation in agricultural practices, thereby contributing to sustainable and resilient farming systems.

6.2 Comparison of Existing and Proposed System

Existing system: (Manual Analysis)

In the existing system, farmers typically rely on traditional farming practices and manual decision-making processes to manage their agricultural operations. These methods often involve drawing on historical knowledge, personal experience, and intuition to make planting decisions, manage resources, and assess crop health. While these approaches may have served farmers adequately in the past, they are inherently limited by subjectivity and reliance on individual expertise. Moreover, traditional methods may struggle to account for the complexities of modern agriculture, such as changing weather patterns, market dynamics, and pest pressures. As a result, farmers may face challenges in optimizing productivity, mitigating risks, and adapting to evolving agricultural conditions. The reliance on manual processes also entails significant time and labor costs, potentially limiting farmers' ability to make informed and timely decisions.

Proposed system: (Random forest algorithm)

The proposed system represents a significant advancement in agricultural decision-making by leveraging ML techniques, specifically a Random Forest algorithm, to predict crop yield. By integrating historical crop data, weather information, and other relevant factors, the system provides farmers with data-driven insights to optimize resource allocation, mitigate risks, and improve productivity. Unlike traditional methods reliant on subjective judgment and historical knowledge, the proposed system offers objective and accurate predictions based on empirical data analysis. Moreover, the system's scalability and adaptability enable it to accommodate the complexities of modern agriculture, such as changing weather patterns and market dynamics. With its user-friendly interface, the system facilitates seamless interaction for farmers of varying technical backgrounds, streamlining the process of inputting data and obtaining predictions. Overall, the proposed system offers a promising solution to enhance agricultural practices, empower farmers, and drive sustainable and resilient farming systems in the face of evolving challenges.

Model	MSE	RMSE	R2 Score
Random Forest	0.29593	0.54400	0.96919
XG Boost	0.30926	0.54918	0.97301
Light GBM	0.31459	0.56088	0.96725

Table 6.1: **Comparison of Different Models in Proposed System**

6.3 Sample Code

```python
# importing all necessary python libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
# Display basic information about the dataset
print("Dataset information:")
data.info()
# Display the first few five of the dataset
print("\nFirst few rows of the dataset:")
data.head()
# Display the last five rows of the dataset
print("\nLast few rows of the dataset:")
data.tail()
# Perform one-hot encoding
encoded_data = pd.get_dummies(data, columns=categorical_columns)
encoded_data.head()
# Split data into features and target variable
X = encoded_data.drop(['Production_in_tons'], axis=1)  # Features
y = encoded_data['Production_in_tons']  # Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions and evaluate
rf_y_pred = rf_model.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_y_pred)
rf_r2 = r2_score(y_test, rf_y_pred)
rf_rmse = np.sqrt(rf_mse)
print("Random Forest:")
print("Mean Squared Error:", rf_mse)
print("R-squared:", rf_r2)
print("Root Mean Squared Error:", rf_rmse)
log_transformed_data = data.copy()
log_transformed_data[numerical_attributes] = log_transformed_data[numerical_attributes].apply(lambda
    x: np.log1p(x))
log_transformed_data.head()
```

Output

```
Please enter the values for each feature:
N: 80
P: 40
K: 20
pH: 5.62
rainfall: 654.34
temperature: 29.266667
Area_in_hectares: 2800.0
NPK_Ratio: 1.3333333
Temp_Rainfall_Interaction: 19150.350667
Crop: maize
State_Name: andhra pradesh
Crop_Type: kharif
Predicted crop yield: 2.755817266498499
```

Figure 6.1: **Crop Yield Prediction**

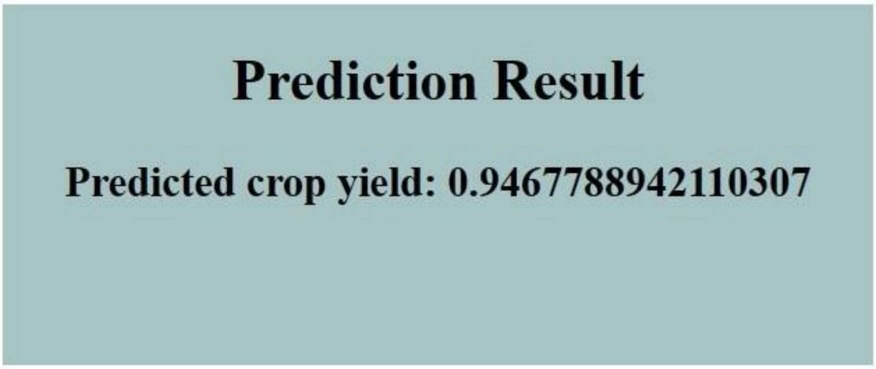

Figure 6.2: **Prediction of Crop Yield through GUI**

The figures 6.1 and 6.2 presents two visual representations one from Google Colab and the other from a GUI-based website. These visuals showcase the model's accuracy in predicting crop yields, leveraging comprehensive datasets and advanced algorithms. The Google Colab output demonstrates robust analysis capabilities, while the website GUI offers user-friendly access, ensuring stakeholders can easily utilize the model's insights for informed decision-making in agriculture.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The development and implementation of the proposed ML based system for crop yield prediction signify a significant step forward in modernizing agricultural practices. Through the utilization of advanced techniques such as the RF algorithm, coupled with comprehensive datasets encompassing historical crop data, weather patterns, and other pertinent variables, the system empowers farmers with actionable insights to optimize their farming operations. By transitioning from subjective decision-making processes to data-driven approaches, farmers can make more informed choices regarding resource allocation, crop management strategies, and risk mitigation, ultimately enhancing productivity and profitability in agricultural endeavors.

The scalability and adaptability of the proposed system make it well-suited to address the challenges posed by modern agriculture, including fluctuating weather patterns, evolving market dynamics, and changing environmental conditions. The system's ability to analyze large volumes of data efficiently enables farmers to adapt their practices dynamically, ensuring resilience and sustainability in agricultural operations. Moreover, the user-friendly interface of the system enhances accessibility for farmers of varying technical backgrounds, facilitating seamless integration into existing farming workflows. Overall, the proposed system represents a promising avenue for leveraging technology to drive innovation in agriculture, fostering sustainable practices, and contributing to the advancement of food security and rural livelihoods in a rapidly changing world.

7.2 Future Enhancements

Several future enhancements could further augment the capabilities and impact of this project. Firstly, integrating real-time data streams into the predictive model would enhance its accuracy and timeliness. By incorporating up-to-date information on weather patterns, soil moisture levels, and pest outbreaks, the system could provide farmers with dynamic, actionable insights to adapt their farming practices in response to changing conditions. Additionally, leveraging remote sensing technologies such as satellite imagery or drones could offer valuable data on crop health, growth stages, and field conditions, further refining the predictive accuracy of the model. These enhancements would enable farmers to make more informed and proactive decisions, leading to improved yield outcomes and resource efficiency.

Another avenue for future enhancement lies in the incorporation of advanced analytics techniques, such as deep learning or ensemble methods, to enhance the predictive capabilities of the model. These techniques have demonstrated significant potential in capturing complex patterns and relationships within agricultural data, thereby improving the accuracy and robustness of crop yield predictions. Moreover by empowering farmers with data-driven guidance in real-time, these enhancements would facilitate more efficient and sustainable agricultural practices, ultimately contributing to enhanced productivity, profitability, and resilience in the face of evolving agricultural challenges.

Chapter 8

PLAGIARISM REPORT

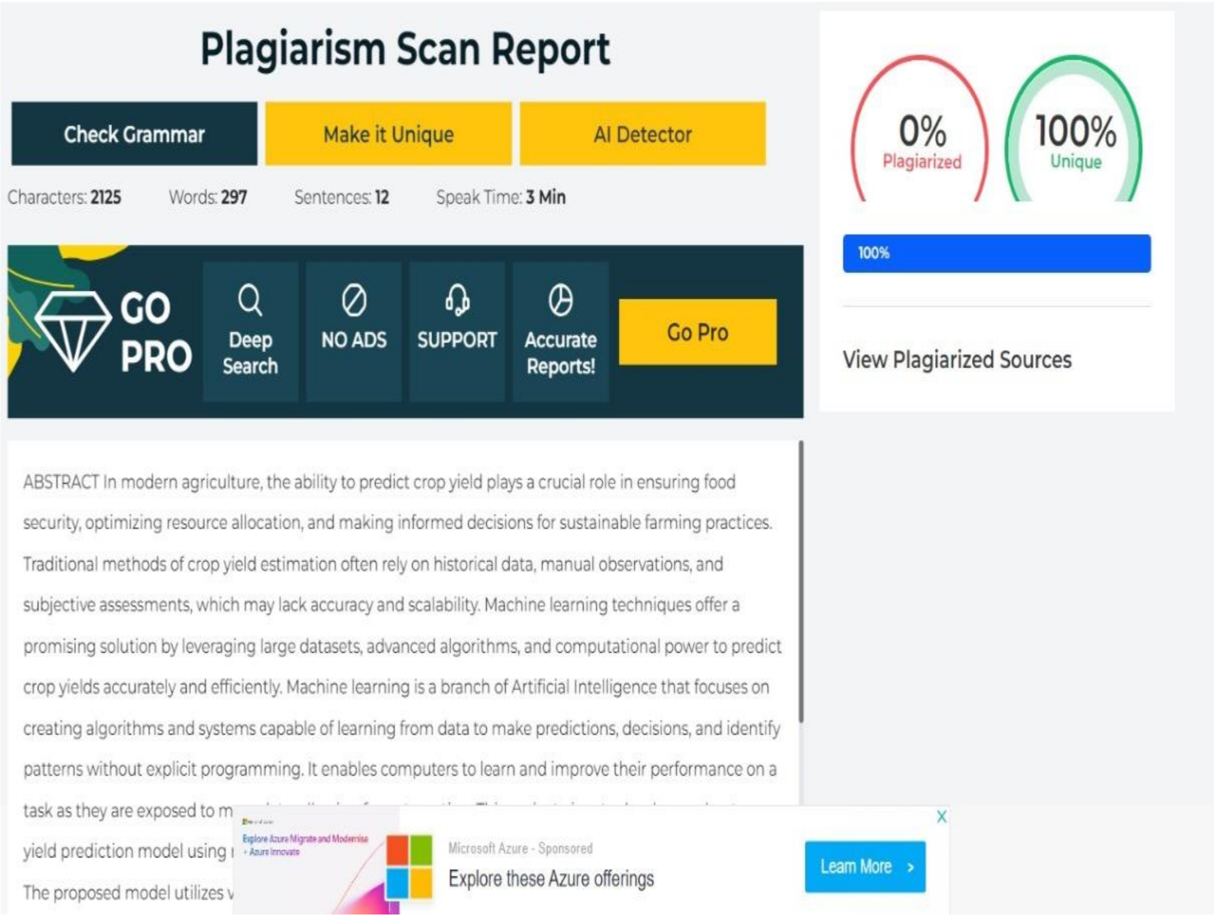

Figure 8.1: **Plagarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
# importing all necessary python libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
# Display basic information about the dataset
print("Dataset information:")
data.info()
# Display the first few five of the dataset
print("\nFirst few rows of the dataset:")
data.head()
# Display the last five rows of the dataset
print("\nLast few rows of the dataset:")
data.tail()
# Display the shape of the dataset
print("\nShape of the dataset:")
print(data.shape)
# Summary statistics for categorical variables
print("\nSummary statistics for categorical variables:")
data.describe(include=['object'])
# Unique values in categorical columns
print("\nUnique values in categorical columns:")
for column in data.select_dtypes(include='object').columns:
    print(f"Column: {column}")
    print(data[column].unique())
# Groupby analysis
print("\nGroupby analysis - Mean production by crop type:")
data.groupby('Crop_Type')['Production_in_tons'].mean()
# Check for missing values
print("\nMissing values:")
data.isnull().sum()
```

```
#Univariate Analysis of attributes
# Numerical attributes for analysis
numerical_attributes = ['N', 'P', 'K', 'pH', 'rainfall', 'temperature', 'Production_in_tons']
# Categorical attributes for analysis
categorical_attributes = ['Crop_Type', 'Crop','State_Name']
# Univariate analysis for numerical attributes
for column in numerical_attributes:
    plt.figure(figsize=(8, 6))
    sns.histplot(data[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()
# Univariate analysis for categorical attributes
for column in categorical_attributes:
    plt.figure(figsize=(16, 8))
    sns.countplot(x=column, data=data)
    plt.title(f'Count of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45)
    plt.show()
# Multivariate Analysis of attributes
numerical_attributes1 = ['N', 'P', 'K', 'pH', 'rainfall', 'temperature', 'Production_in_tons','
    Area_in_hectares','Yield_ton_per_hec']
# Correlation matrix for numerical attributes
correlation_matrix = data[numerical_attributes1].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix for Numerical Attributes')
plt.show()
# Feature Creation
# NPK Ratio
data['NPK_Ratio'] = data['N'] / (data['P'] + data['K'])
# Temperature-Rainfall Interaction
data['Temp_Rainfall_Interaction'] = data['temperature'] * data['rainfall']
# Display the updated dataset with derived features
data.head()
 # Apply logarithmic transformation
log_transformed_data = data.copy()
log_transformed_data[numerical_attributes] = log_transformed_data[numerical_attributes].apply(lambda
     x: np.log1p(x))
log_transformed_data.head()
# Categorical columns to one-hot encode
categorical_columns = ["State_Name", "Crop_Type", "Crop"]
# Perform one-hot encoding
encoded_data1 = pd.get_dummies(log_transformed_data, columns=categorical_columns)
encoded_data1.head()
# Split data into features and target variable
X = encoded_data1.drop(['Production_in_tons','Unnamed: 0','Yield_ton_per_hec'], axis=1)  # Features
```

```python
y = encoded_data1['Production_in_tons']  # Target variable
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train Random Forest model
rf_model1 = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model1.fit(X_train, y_train)
# Make predictions and evaluate
rf_y_pred = rf_model1.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_y_pred)
rf_r2 = r2_score(y_test, rf_y_pred)
rf_rmse = np.sqrt(rf_mse)
print("Random Forest:")
print("Mean Squared Error:", rf_mse)
print("R-squared:", rf_r2)
print("Root Mean Squared Error:", rf_rmse)
# Get the feature names used during model training
feature_names = X.columns.tolist()
# Create a dictionary to store user input for each feature
new_data = {}
# Ask user to enter values for each feature
print("Please enter the values for each feature:")
for feature in feature_names:
    if not feature.startswith(('Crop_', 'Crop_Type_', 'State_Name_')):
        value = float(input(f"{feature}: "))  # Convert input to float if necessary
        # Apply logarithmic transformation if needed
        if feature in ['N', 'P', 'K', 'pH', 'rainfall', 'temperature', 'Area_in_hectares', 'NPK_Ratio', 'Temp_Rainfall_Interaction']:
            value = np.log(value + 1)  # Adding 1 to avoid log(0)
        new_data[feature] = value
# Ask user to enter the categorical variables
crop_name = input("Crop: ")
state_name = input("State_Name: ")
crop_type = input("Crop_Type: ")
# Map the categorical variables to the corresponding one-hot encoded column
crop_column = f"Crop_{crop_name.lower()}"
crop_type_column = f"Crop_Type_{crop_type.lower()}"
state_name_column = f"State_Name_{state_name.lower()}"
# Set the mapped columns to 1
new_data[crop_column] = 1 if crop_column in feature_names else 0
new_data[crop_type_column] = 1 if crop_type_column in feature_names else 0
new_data[state_name_column] = 1 if state_name_column in feature_names else 0
# Set all other one-hot encoded columns to 0
for column in feature_names:
    if column.startswith(('Crop_', 'Crop_Type_', 'State_Name_')):
        if column not in [crop_column, crop_type_column, state_name_column]:
            new_data[column] = 0
# Convert the dictionary into a DataFrame with a single row
new_data_df = pd.DataFrame([new_data])
# Reorder DataFrame columns based on feature names
new_data_df = new_data_df[feature_names]
```

```
# Make predictions on new data
new_predictions = rf_model1.predict(new_data_df)
# Print the predicted crop yield
print("Predicted crop yield:", new_predictions[0])
from joblib import dump
model_filename = "rf_model1.joblib"
dump(rf_model1, model_filename)
print(f"Trained model saved as {model_filename}")
```

9.2 Poster Presentation

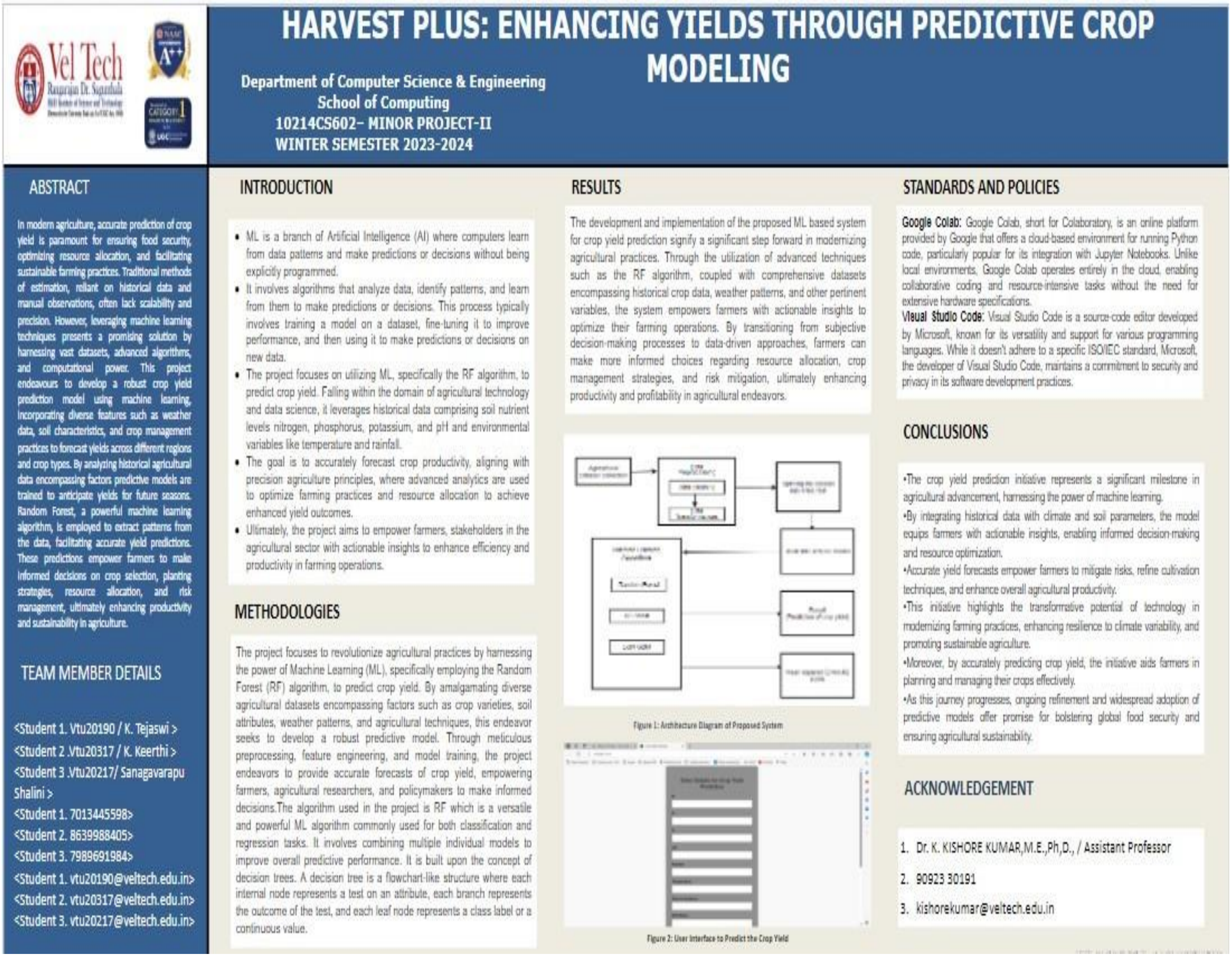

Figure 9.1: **Poster**

References

[1] M. Ashfaq, I. Khan, A. Alzahrani, M. U. Tariq, H. Khan and A. Ghani, ”Accurate Wheat Yield Prediction Using Machine Learning and Climate-NDVI Data Fusion,” in IEEE Access, vol. 12, pp. 40947-40961, 2024.

[2] M. J. Hoque, ”Incorporating Meteorological Data and Pesticide Information to Forecast Crop Yields Using Machine Learning,” in IEEE Access, vol. 12, pp. 47768-47786, 2024.

[3] A. Reyana, S. Kautish, P. M. S. Karthik, I. A. Al-Baltah, M. B. Jasser and A. W. Mohamed, ”Accelerating Crop Yield: Multisensor Data Fusion and Machine Learning for Agriculture Text Classification,” in IEEE Access, vol. 11, pp. 20795-20805, 2023.

[4] A. Sharma, A. Jain, P. Gupta and V. Chowdary, ”Machine Learning Applications for Precision Agriculture: A Comprehensive Review,” in IEEE Access, vol. 9, pp. 4843-4873, 2021.

[5] U. Shafi et al., ”Tackling Food Insecurity Using Remote Sensing and Machine Learning-Based Crop Yield Prediction,” in IEEE Access, vol. 11, pp. 108640-108657, 2023.

[6] H. R. Seireg, Y. M. K. Omar, F. E. A. El-Samie, A. S. El-Fishawy and A. Elmahalawy, ”Ensemble Machine Learning Techniques Using Computer Simulation Data for Wild Blueberry Yield Prediction,” in IEEE Access, vol. 10, pp. 64671-64687, 2022.

[7] S. P. Raja, B. Sawicka, Z. Stamenkovic and G. Mariammal, ”Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers,” in IEEE Access, vol. 10, pp. 23625-23641, 2022.

[8] P. Sharma, P. Dadheech, N. Aneja and S. Aneja, ”Predicting Agriculture Yields Based on Machine Learning Using Regression and Deep Learning,” in IEEE Access, vol. 11, pp. 111255-111264, 2023.

[9] M. Rashid, B. S. Bari, Y. Yusup, M. A. Kamaruddin and N. Khan, ”A Comprehensive Review of Crop Yield Prediction Using Machine Learning Approaches

With Special Emphasis on Palm Oil Yield Prediction," in IEEE Access, vol. 9, pp. 63406-63439, 2021.

[10] Y. Alebele et al., "Estimation of Crop Yield From Combined Optical and SAR Imagery Using Gaussian Kernel Regression," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10520-10534, 2021.