AGRI-SMART ADVISOR: HARVESTIFY

A PROJECT REPORT

Submitted by

BOOBASH AYYANAR.M -RA2211028020021
HARISH RAJ.M -RA2211028020025
KARTHIK.R -RA2211028020026
SRIRAM.K -RA2211028020038

Under the guidance of

Dr. J. Jean Justus

(Associate Professor, Department of Computer Science and Engineering)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

WITH SPECIALIZATION IN CLOUD COMPUTING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI -600089

NOV 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"AGRI-SMART ADVISOR: HARVESTIFY"** is the bonafide work of **M. BOOBASH AYYANAR [REG NO: RA2211028020021], M. HARISH RAJ [REG NO: RA2211028020025], R. KARTHIK [REG NO: RA2211028020026], K. SRIRAM [REG NO: RA2211028020038]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE

Dr. J. Jean Justus

Associate Professor

Computer Science and Engineering,
SRM Institute of Science and Technology,

Ramapuram, Chennai.

SIGNATURE

Dr. K. RAJA, M.E., Ph.D.,

Professor and Head

Computer Science and Engineering,
SRM Institute of Science and Technology,

Ramapuram, Chennai.

Submitted for the project viva-voce held on____________ at

SRM Institute of Science and Technology, Ramapuram, Chennai -600089.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM, CHENNAI - 89

DECLARATION

We hereby declare that the entire work contained in this project report titled "**AGRI-SMART ADVISOR: HARVESTIFY**" has been carried out by **M.BOOBASH AYYANAR [REG NO: RA2211028020021], M.HARISH RAJ [REG NO: RA2211028020025], R.KARTHIK [REG NO: RA2211028020026], K.SRIRAM [REG NO: RA2211028020038]** at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **Dr. J. Jean Justus, Associate professor**, Department of Computer Science and Engineering.

Place: Chennai **BOOBASH AYYANAR.M**

Date: **HARISH RAJ.M**

KARTHIK.R

SRIRAM.K

ABSTRACT

Agriculture is vital to India's economy, with most people depending on it for their livelihoods. Harvestify is a smart, web-based application designed to revolutionize agricultural practices by providing tailored recommendations and predictions using advanced machine learning and deep learning techniques. The platform offers three core functionalities: Crop Recommendation, Fertilizer Suggestion, and Plant Disease Detection. The Crop Recommendation system uses machine learning algorithms to analyze soil composition, weather conditions, and other environmental factors to recommend the most suitable crops for cultivation. By integrating real-time weather data via APIs and IoT sensor inputs for soil monitoring, it offers precise, data-driven insights to farmers. The Fertilizer Suggestion feature analyzes soil nutrient levels and the crop's specific requirements to recommend optimized fertilizer usage, ensuring balanced nutrient supplementation for better yields. Additionally, the Plant Disease Detection module employs a deep learning-based image classification model to diagnose plant diseases from leaf images, enabling early detection and timely treatment recommendations. Harvestify aims to improve agricultural efficiency, reduce resource wastage, and increase crop productivity, ultimately supporting sustainable farming practices and aiding farmers with critical decision-making tools. By leveraging machine learning models, these applications offer tailored recommendations based on user-provided soil data and plant images, enabling farmers to make informed decisions about crop selection, soil management, and disease control.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

ML -	Machine Learning
DL-	Deep Learning
N-	Nitrogen
P-	Phosphorus
K-	Potassium
NB -	Naïve Bayes
RF-	Random Forest
SVM-	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 Introduction

The agriculture stands as a pivotal sector in shaping India's economic landscape, sustaining the livelihoods of a significant portion of its population. With the advent of modern technologies, such as Machine Learning (ML) and Deep Learning (DL), integrated into agricultural practices, the potential for enhancing productivity and sustainability in farming has surged. Nowadays agriculture has developed a lot in India. “site-specific” farming is the key to Precision agriculture. Although precision agriculture has achieved better enhancements it is still facing certain issues. Precision agriculture plays an important role in the recommendation of crops. Harvestify emerges as a pioneering initiative, embodying the fusion of ML and DL methodologies to provide holistic solutions to farmers. Driven by the imperative to optimize agricultural yield and mitigate risks, Harvestify encompasses three core applications: Crop Recommendation, Fertilizer Recommendation, and Disease Detection. These applications are designed to empower farmers with data-driven insights, aiding decision-making processes crucial for crop management. This project introduces a proof-of-concept website designed to leverage ML and DL technologies to enhance precision farming. The goal of this project is to provide farmers with intelligent tools that can assist them in making data-driven decisions to improve crop management, optimize resource usage, and increase overall productivity. The website encompasses three primary functionalities: crop recommendation, fertilizer recommendation, and plant disease detection. This paper aims to recommend the most suitable crop based on input parameters like Nitrogen (N), Phosphorous (P), Potassium (K), PH value of soil, Humidity, Temperature, and Rainfall. This paper predicts the accuracy of the future production of eleven different crops such as rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mungbean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, and coffee crops using various supervised machine learning approaches in of India and recommends the most suitable crop. The dataset contains various parameters like Nitrogen (N), Phosphorous (P), Potassium (K), PH value of soil, Humidity, Temperature, and Rainfall. Machine Learning focuses on the algorithm like supervised, unsupervised, and Reinforcement learning and each of them has its advantages and disadvantages. Supervised learning the algorithm assembles a mathematical model from a set of data that contains both the inputs and the desired outputs. An unsupervised learning-the algorithm constructs a mathematical model from a set of data that contains only inputs and no desired output labels. Semi-supervised learning- algorithms expand mathematical models from incomplete training data, where a portion of the sample input doesn't have labels .This proposed system applied different kinds of Machine Learning algorithms like Decision Trees, Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression, Random Forest (RF), and XGBoost. The goal of this project is to provide farmers with intelligent tools that can assist them in making data-driven decisions to improve crop management, optimize resource usage, and increase overall productivity. The website encompasses three primary functionalities: crop recommendation, fertilizer recommendation, and plant disease detection.

1.2 Problem Statement

The agricultural sector faces declining crop yields, inefficient fertilizer use, and increasing plant diseases, worsened by climate change and limited access to timely information. This results in suboptimal decision-making, highlighting the need for a data-driven solution to support farmers in enhancing productivity and sustainability.

1.3 Objective of the Project

Harvestify aims to develop a web application that employs Machine Learning and Deep Learning to provide personalized crop recommendations, optimize fertilizer application, and predict plant diseases. The goal is to deliver real-time insights that empower farmers to improve their agricultural practices.

1.4 Project Domain

Harvestify operates in the domain of Precision Agriculture, which combines agricultural science with technology and data analytics to optimize farming practices, enhance resource efficiency, and improve crop management while minimizing environmental impact.

1.5 Scope of the Project

The project includes developing a user-friendly web interface, integrating data from satellite imagery, weather forecasts, and soil metrics, and utilizing various Machine Learning algorithms for accurate predictions. Harvestify is designed to be scalable and adaptable to diverse agricultural contexts.

1.6 Methodology

The methodology involves data collection and preprocessing, followed by the selection and training of Machine Learning models. The refined models will be integrated into a web application, providing farmers with actionable insights. User feedback will be gathered for continuous improvement of the application. In our framework, we have proposed a procedure that is separated into various stages as appeared in The five phases are as per the following:Collection of Datasets ,Pre-processing (Noise Removal),Feature Extraction ,Applied Various Machine Learning Algorithm ,Recommendation System

CHAPTER 2

LITERATURE REVIEW

Paper [1] explains that agriculture is key to India's economy, with many relying on it as their primary occupation. Poor crop quality often stems from improper fertilizer use, either excessive or insufficient. An IoT and ML system for soil testing addresses this issue by using sensors to monitor soil parameters like temperature, moisture, pH, and NPK levels. Data from these sensors is analyzed using ML algorithms like random forest to suggest optimal crop growth conditions. By applied this proposed system, we can predict best crop according to the field weather conditions. This crop prediction can be done by random forest algorithm and decision tree. An ML system for soil testing addresses this issue by using sensors to monitor soil parameters like temperature, moisture, pH, and NPK levels. Data from these sensors is analyzed using ML algorithms like random forest to suggest optimal crop growth conditions. Paper [2] proposed an algorithm that achieved 71% accuracy by using rainfall predictor model and achieved 91.00% accuracy by applying neural network algorithm on crop suitable predictor system The desktop application takes in nutrient values like pH, Nitrogen (N), Phosphorus (P), and Potassium (K) from the soil. The application recommends fertilizers to use to make crops grow better. Paper [3] is used to identify particular crop according to given particular data. By applied this proposed system, we can predict best crop according to the field weather conditions. This crop prediction can be done by random forest algorithm and decision tree. By applying random forest algorithm got best accurate value result. More accuracy results gave more profit to the crop yield. By applying Support Vector Machine (SVM) acquired higher precision and productivity.

Paper [4] explains various machine learning classifiers like support vector machine (SVM) classifier, ANN classifier, Random Forest and Naïve Bayes for recommend a crop for site specific parameter with accuracy and efficiency This research work would help farmers to increase productivity in agriculture, prevent soil degradation in cultivated land, and reduce chemical use in crop production and efficient use of water resources. By applying this proposed system achieved 99.91% accuracy result. Paper [5] proposed a framework that gives simple openness to the clients. They are additionally simple to utilize and comprehend by the trifler ranchers. It improves the perception and Understandability. By using this proposed system recommended particular crop according to their Nutrients (N, P, K, and PH) values and also identified available Nutrients values and required fertilizers quantities for the particular crop like Rice, Maize, Black gram, Carrot and Radish. Paper [6] helps to the impetus for this research stems from the increasing acceptance of self-managed farming practices as well as the need for accurate, unbiased analysis and farming action adjustment. The impetus for this research stems from the increasing acceptance of self-managed farming practices as well as the need for accurate, unbiased analysis and farming action adjustment. This proposed system achieved 71% accuracy by using rainfall predictor model and achieved 91.00% accuracy by applying neural network algorithm on crop suitable predictor system.

CHAPTER 3

PROJECT DESCRIPTION

3.1 Existing System

The existing agricultural practices largely rely on traditional methods where farmers receive generalized recommendations for crop selection and fertilizer use. These systems do not consider real-time farm data or environmental factors, making them less precise. Manual plant disease detection is another common approach, which often results in delays in diagnosis and treatment, leading to potential crop loss. Key challenges include a lack of tailored recommendations, reliance on manual inspection for disease detection, and minimal use of real-time data for decision-making. This leads to inefficiencies, such as improper fertilizer application, delayed responses to disease outbreaks, and suboptimal crop yields.

3.2 Proposed System

The Harvestify offers an innovative solution that integrates Machine Learning (ML), Deep Learning (DL), and IoT technologies to revolutionize farming practices. By leveraging ML models, the system provides precise crop recommendations based on soil type, weather conditions, and historical patterns, allowing farmers to make data-driven decisions that optimize crop yield and sustainability. The fertilizer recommendation module analyzes soil and crop needs to suggest the ideal type and quantity of fertilizer, minimizing waste and reducing environmental impact. Additionally, the plant disease prediction module uses advanced image processing techniques, employing deep learning models to automatically detect diseases from plant images in real-time. Harvestify stands out by offering personalized, real-time, and automated solutions that empower farmers to enhance productivity and make more informed decisions.

3.2.1 Advantages

- **Real-Time Data Integration**: By incorporating IoT sensors, the system collects real-time environmental data such as soil moisture, temperature, and humidity to make more accurate recommendations.
- **Personalized Solutions**: Unlike traditional systems, Harvestify tailors its recommendations based on specific farm conditions, leading to better decision-making.
- **Early Disease Detection**: The use of deep learning models for image-based disease detection helps farmers address issues early, reducing crop damage and improving yields.

3.3 Feasibility Study

A feasibility study is crucial to determine whether the proposed Harvestify system can be successfully implemented from economic, technical, and social perspectives. Each aspect is explored below:

3.3.1 Economic Feasibility

The economic feasibility of Harvestify is strong, as the system offers significant cost-saving opportunities for farmers. By providing accurate crop and fertilizer recommendations, it minimizes wasteful fertilizer use and reduces costs associated with over-application. The early detection of plant diseases allows farmers to take timely actions, preventing large-scale crop losses and reducing the need for expensive interventions later. The initial investment in deploying IoT sensors and adopting the platform can be offset by long-term savings in inputs, increased yields, and reduced labor costs associated with manual disease monitoring and decision-making. Moreover, the scalable nature of the platform means that it can be adapted to various farm sizes and types, ensuring affordability for small-scale farmers as well as large commercial operations

3.3.2 Technical Feasibility

From a technical standpoint, Harvestify is feasible due to the widespread availability of the technologies it integrates. The system leverages well-established Machine Learning (ML) and Deep Learning (DL) algorithms, which can be implemented efficiently using existing cloud-based services and modern programming frameworks. The integration of IoT sensors for real-time data collection is also highly achievable with current sensor technology, which is affordable and widely accessible. Additionally, the use of cloud computing infrastructure allows for scalable and flexible deployment, accommodating varying amounts of data and user needs. The technical skills required to develop and maintain the system, such as expertise in machine learning, data science, and web development, are well-established and available, ensuring that the system can be built and maintained effectively.

3.3.3 Social Feasibility

Harvestify is socially feasible, as it addresses key challenges faced by farmers and aligns with broader societal goals of sustainable agriculture and food security. By offering personalized, data-driven recommendations, the system empowers farmers to make informed decisions, improving productivity and reducing environmental impact. The use of IoT and AI technologies is increasingly accepted in agriculture, and many farmers are open to adopting innovative solutions that can simplify their operations and improve yields. Additionally, Harvestify' s ability to support multiple languages and provide localized recommendations ensures that it can be adopted by farmers from diverse regions and backgrounds. The platform's focus on sustainability also makes it socially responsible, contributing to the global goal of reducing the environmental footprint of farming while enhancing food production.

3.4 System Specification

3.4.1 Hardware Specification

- Processor- 12th Gen Intel(R) Core (TM) i5-1235U 1.30 GHz
- RAM-8GB
- OS- 64-bit operating system, x64-based processor
- Windows 11

3.4.2 Software specification

- Python
- Python modules
- Anaconda
- Git
- Vs code/notepad+

CHAPTER 4

PROPOSED METHODOLOGY

4.1 General Architecture

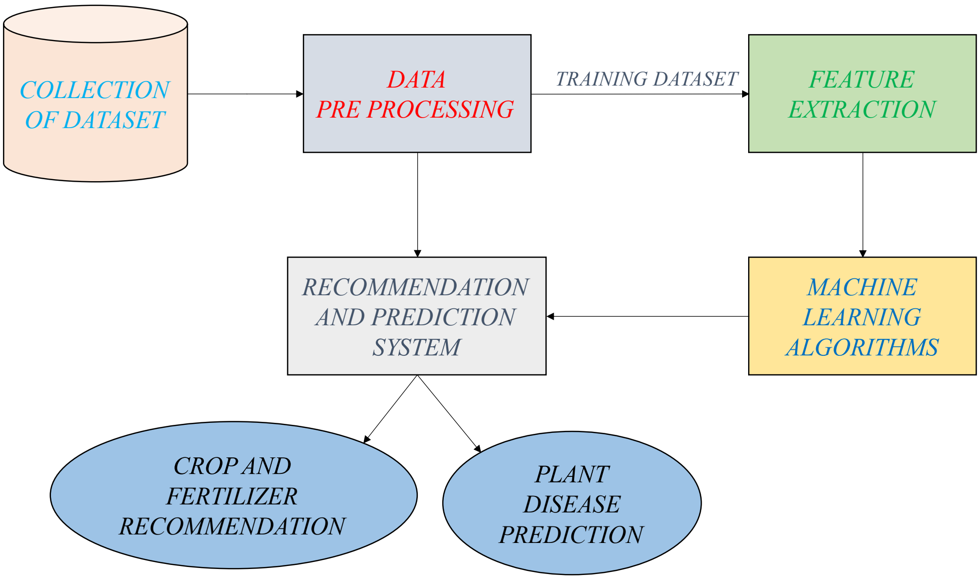

Figure 4.1: **Architecture Diagram**

Figure 4.1 outlines the general workflow of a recommendation and prediction system. It starts with the collection of relevant datasets, which are then divided into training and testing sets to evaluate the model's performance. Feature extraction involves identifying and selecting the most important characteristics of the data that will be used for prediction. These features are then fed into different machine learning algorithms to train the model. The trained model is then tested on the unseen testing data to assess its accuracy. Finally, based on the model's predictions and other factors like soil conditions, climate, and market demand, the system recommends the most suitable crop for a given region.

4.2 Design Phase

4.2.1 Block Diagram For Crop Recommendation and Fertilizer Recommendation

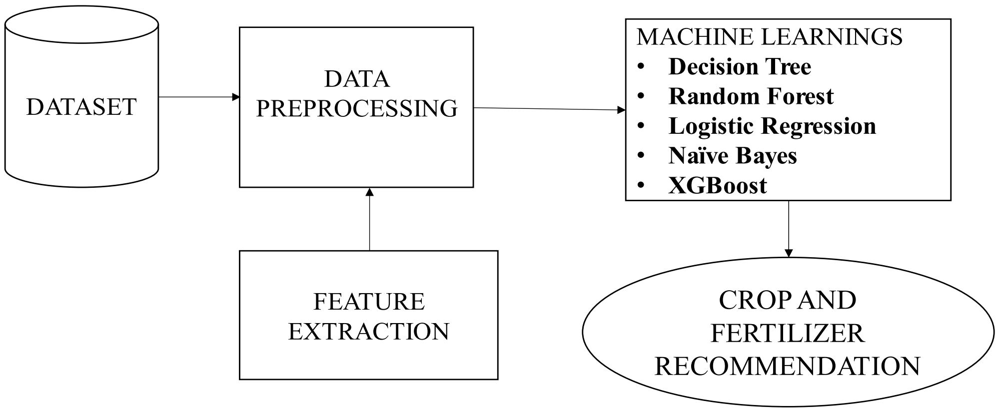

Figure 4.2: **Block Diagram For Crop Recommendation and Fertilizer Recommendation**

Figure 4.2 illustrates a crop and fertilizer recommendation system that utilizes machine learning techniques. It starts by collecting data from various sources, including public repositories like Kaggle and UCI, as well as external sources like climate and soil data. Once collected, the data undergoes a preprocessing phase to ensure its quality and consistency. This preprocessed data is then fed into machine learning algorithms, which are trained to predict suitable crops and fertilizers based on the specific characteristics of the region. The system's output includes recommendations for the most appropriate crop and fertilizer choices, providing valuable guidance to farmers in optimizing their agricultural practices.

4.2.2 Block Diagram For Plant Disease Prediction

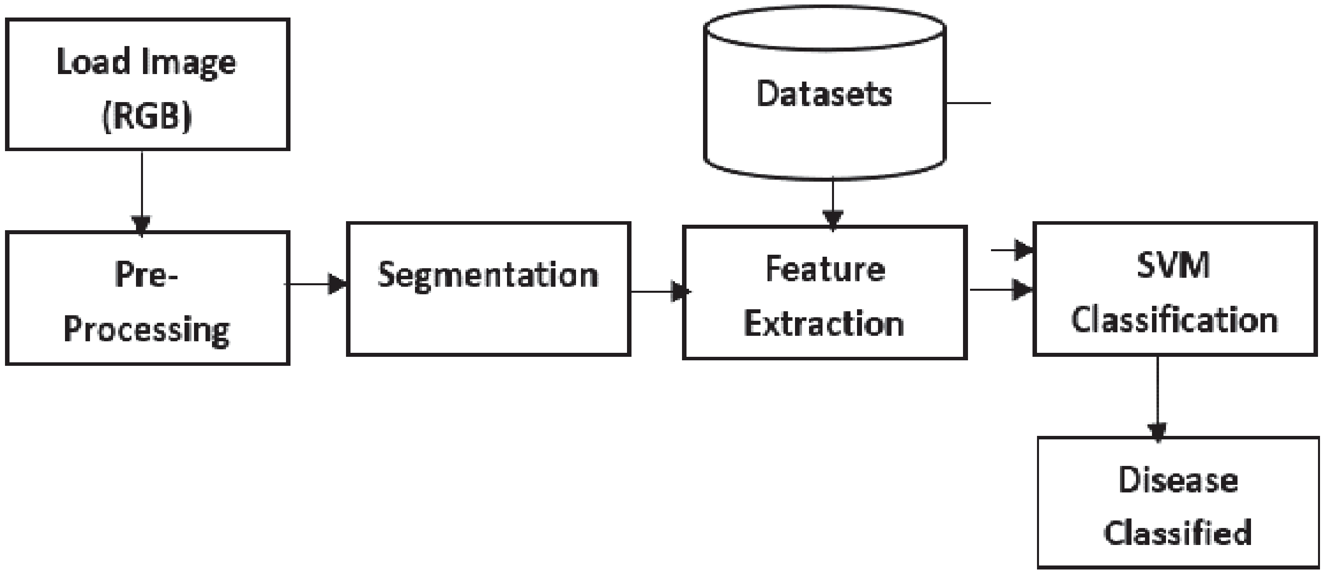

Figure 4.3: **Block Diagram for Plant Disease Prediction**

Figure 4.3 illustrates the process of plant disease classification using image analysis. An RGB image is loaded and pre-processed to improve quality, followed by segmentation to isolate the affected area. Features are then extracted and, along with datasets, fed into an SVM classifier, which classifies the disease based on the extracted features.It starts by loading an RGB image of the affected plant. The image then undergoes preprocessing to enhance its quality and remove noise. Next, the image is segmented to identify the regions of interest, which are likely the diseased areas. Feature extraction involves extracting relevant characteristics from these regions, such as color, texture, and shape. Finally, these extracted features are fed into a Support Vector Machine (SVM) classifier, which is trained to distinguish between different plant diseases. The SVM model then predicts the disease based on the extracted features, providing a valuable tool for farmers in diagnosing and managing plant health issues

4.3 PROPOSED SYSTEM

In our framework, we have proposed a procedure that is separated into various stages as appeared in The five phases are as per the following:

4.3.1 Collection of Datasets

The first step in the Harvestify project is the collection of datasets, which is crucial for training machine learning models and generating accurate recommendations. Various data sources are utilized to gather relevant information, including agricultural databases that contain historical records of soil characteristics, crop yields, and climatic conditions. Additionally, real-time data is collected from IoT sensors deployed in the field, measuring parameters such as soil moisture, temperature, humidity, and nutrient levels. User inputs, including specific crop types, soil test results, and environmental conditions, are also incorporated to enrich the dataset. Furthermore, external weather APIs provide real-time weather information and forecasts, ensuring that the models have access to the most current climatic data. This comprehensive dataset forms the foundation for subsequent analysis and model training.

4.3.2 Pre-processing

The next step is preprocessing, which focuses on cleaning the data to ensure its quality and suitability for analysis. This phase involves handling missing values by employing techniques such as imputation, where missing data points are substituted with mean or mode values, or by removing records with substantial gaps. Noise removal is another critical aspect, as it entails filtering out irrelevant or erroneous data points that could adversely affect model performance. This includes identifying and eliminating outliers and duplicate entries. By meticulously cleaning the data, the preprocessing stage ensures that only high-quality, reliable information.

4.3.3 Feature Extraction

Feature extraction is a vital step in preparing the data for machine learning models. In this phase, key attributes or variables that significantly contribute to predictive performance are identified and selected from the preprocessed datasets. This process involves analyzing the relationships between various factors, such as soil properties, climatic conditions, and crop types, to determine which features are most relevant for accurate predictions. Techniques such as correlation analysis, principal component analysis (PCA), and domain knowledge are utilized to create a refined set of features that enhances the model's ability to learn patterns and make predictions. By focusing on the most impactful features, this step improves the efficiency and effectiveness of the subsequent machine learning algorithms.

4.3.4 Machine Learning Algorithms

After feature extraction, the Harvestify project applies various machine learning algorithms to train models for crop recommendations, fertilizer suggestions, and plant disease predictions. Different algorithms, including Decision Trees, Random Forest, Support Vector Machines, and XGBoost, are employed to analyze the extracted features and learn from the datasets. Each algorithm offers unique strengths; for instance, Random Forest is known for its robustness against overfitting, while XGBoost excels in handling large datasets with high-dimensional features. Through training and validation, the models are optimized to ensure high accuracy and reliability in their predictions.

The algorithms use are following:

1. **Crop Recommendation system:**

- Decision Tree: Builds a tree-like model for crop prediction, though prone to overfitting.
- Random Forest: Combines multiple trees for robust and accurate predictions.
- SVM: Finds optimal boundaries for crop classification.
- XGBoost: Handles large datasets to predict crops with high accuracy using a boosting approach.

2. **Fertilizer Recommendation System:**

- Logistic Regression: Simple model predicting fertilizer needs based on soil data.
- Naïve Bayes: Fast, assumes feature independence for efficient recommendations.
- XGBoost: Pinpoints nutrient deficiencies with high accuracy.

3. **Plant Disease Prediction:**

- Random Forest: Robust disease identification through tree ensemble learning.
- SVM: Classifies diseases by finding separating hyperplanes.
- XGBoost: Strong classifier for disease detection based on image features.

4.3.5 Recommendation System

The final stage of the pipeline is the development of a recommendation system that synthesizes the outputs from the trained machine learning models to provide tailored insights to farmers. This system takes the predictions generated by the crop recommendation, fertilizer recommendation, and plant disease prediction models and presents them in an accessible format through the user interface. Farmers can input their specific data and receive personalized recommendations that guide them in making informed decisions about crop selection, fertilizer application, and disease management. By integrating real-time data and predictive analytics, the recommendation system empowers farmers to optimize resource utilization, enhance crop productivity, and adopt sustainable agricultural practices, ultimately contributing to improved agricultural outcomes.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

In the Harvestify project, the input and output processes for each application are tailored to provide actionable insights based on user data. In the Crop Recommendation Application, users enter various soil parameters, including pH, moisture, nitrogen, phosphorus, and potassium levels. The system processes this information and predicts which crops are best suited to the soil conditions, helping farmers choose crops that will maximize yield. For the Fertilizer Recommendation Application, users input soil data along with the crop they are cultivating. The system then analyzes the soil's nutrient profile and recommends specific fertilizers or soil amendments to correct deficiencies or excesses, ensuring the soil is balanced for optimal crop growth. Lastly, in the Plant Disease Prediction Application, users upload images of diseased plant leaves. The model examines the images, identifies the likely disease, provides a brief explanation of the disease, and suggests treatment options. This helps farmers quickly address plant health issues, preventing crop loss and improving farm productivity.

5.1.1 Crop Recommendation System

In the crop recommendation application, the user can provide the soil data from their side and the application will predict which crop should the user grow. In this Application, users provide soil data (such as pH, nitrogen, potassium levels, etc.), and the application analyzes this data to predict the most suitable crops to grow in those soil conditions. The system uses machine learning models to identify patterns in the soil data and match them to crops that are most likely to thrive, improving farming decisions and yields.

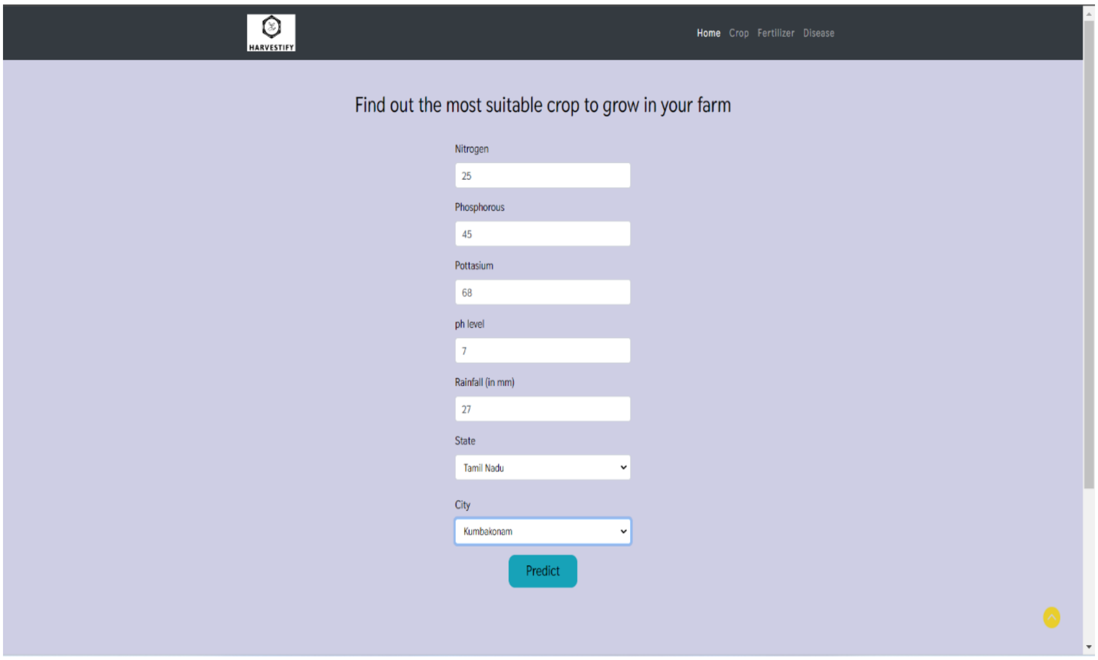

Figure 5.1: **Crop Recommendation Input**

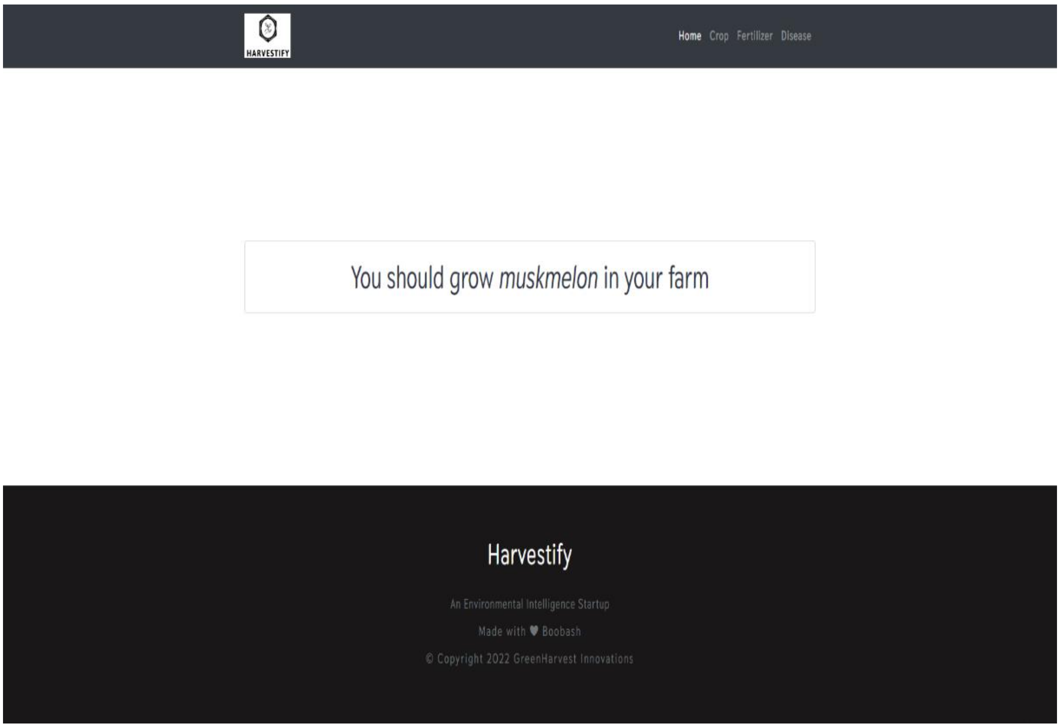

Figure 5.2: **Crop Recommendation Output**

5.1.2 Fertilizer Recommendation

In the Fertilizer Recommendation Application, users input both soil data and the type of crop they are growing. The application evaluates the nutrient levels and other soil properties to determine if the soil has deficiencies or excesses in essential nutrients like nitrogen or phosphorus. Based on this analysis, the system recommends specific fertilizers or soil improvements to optimize crop growth, ensuring the soil is balanced for the crop.

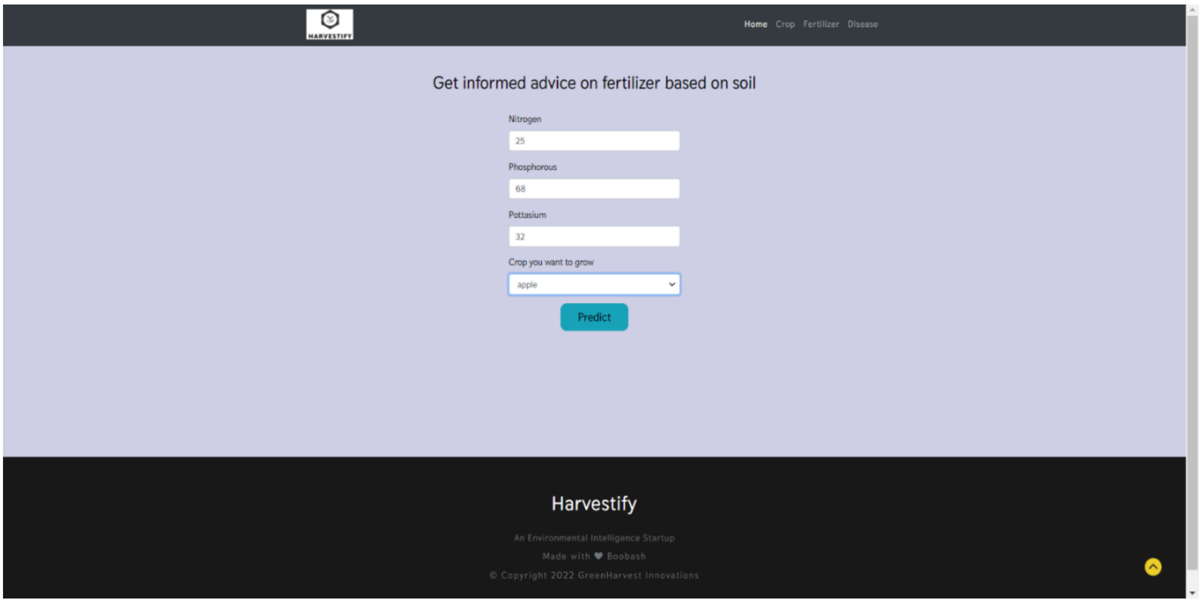

Figure 5.3: **Fertilizer Recommendation Input**

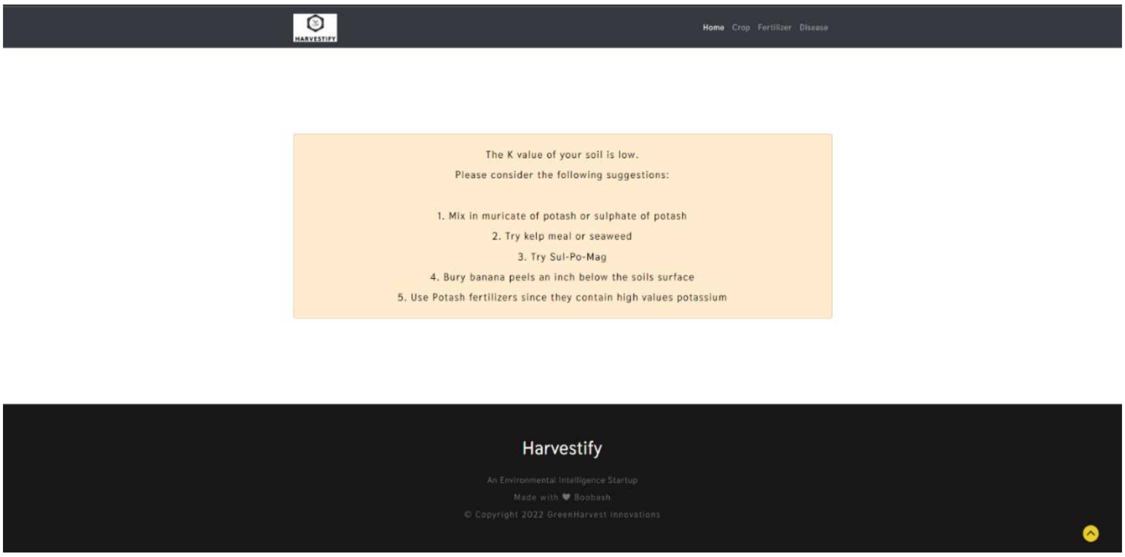

Figure 5.4: **Fertilizer Recommendation Output**

5.1.3 Plant Disease Prediction

The Plant Disease Prediction Application allows users to upload an image of a diseased plant leaf. The application processes the image using machine learning and image recognition techniques to identify the plant disease. Once a disease is identified, the system provides a brief description of the disease and offers suggestions for treatment, helping farmers quickly address plant health issues and prevent crop damage.

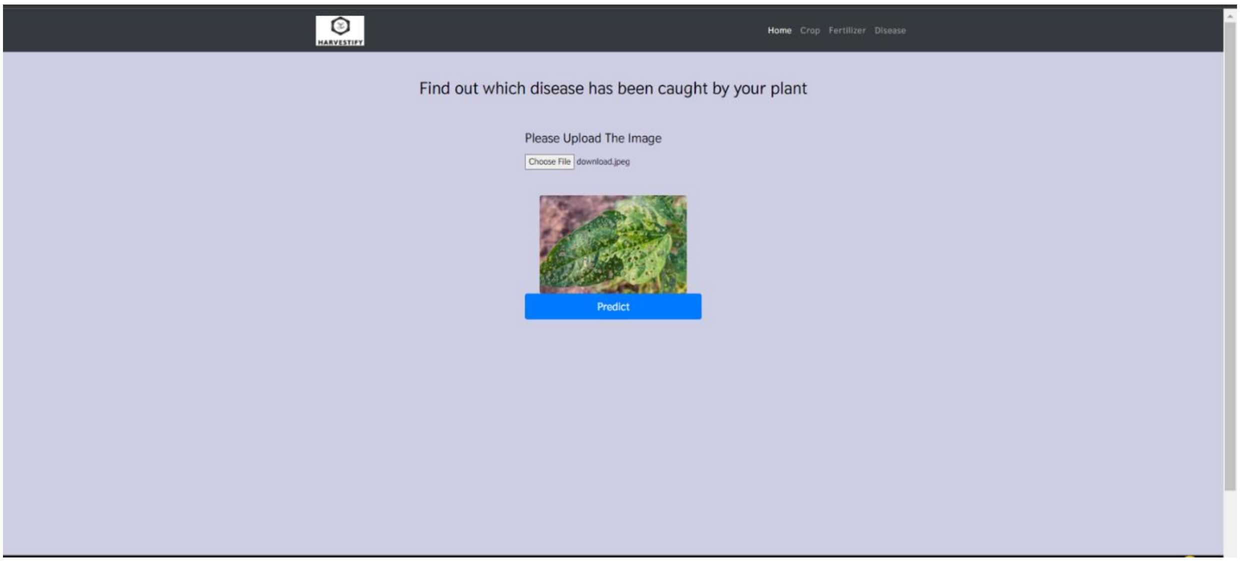

Figure 5.5: **Disease Prediction Input**

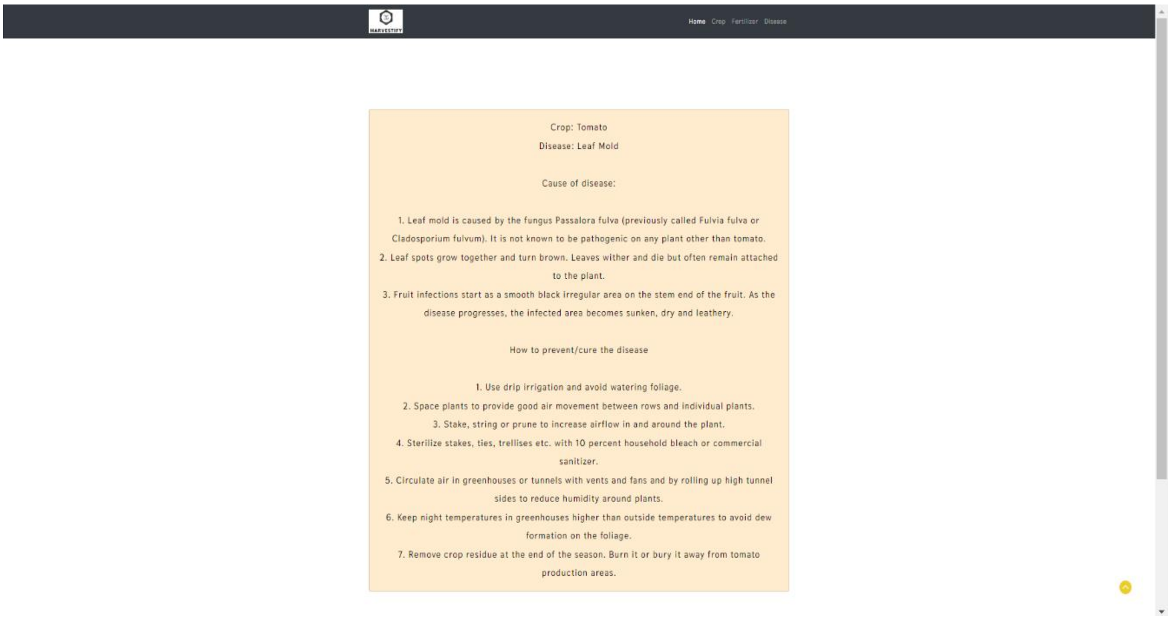

Figure 5.6: **Disease Prediction Output**

5.2 Testing

Testing is a crucial phase in the development of the Harvestify project, ensuring that all components of the system work as intended and meet the required specifications.

5.2.1 Unit testing

Unit testing focuses on individual units or components of the Harvestify application, such as functions, classes, or modules. Automated tests will be written using testing frameworks like PyTest or unittest in Python. Each function will be tested for correct output, handling of edge cases, and proper error handling. This process helps identify bugs early in the development cycle, ensuring that each component functions as intended before integration.

5.2.2 Integration testing

Integration testing involves testing the interactions between different modules of the Harvestify application. This testing ensures that data flows correctly between components, such as between the front-end interface and back-end machine learning models, as well as between the IoT sensors and the data processing system. Automated integration tests will be written to validate that the integrated system performs correctly and meets the expected behavior, particularly in handling data inputs and outputs across modules.

5.2.3 Functional testing

Functional testing evaluates the Harvestify application against its functional requirements. Test cases will be developed based on user stories and specifications, focusing on scenarios such as user registration, crop recommendations, fertilizer suggestions, and disease prediction. Both manual and automated testing approaches will be employed, using tools like Selenium for web application testing, to ensure that each function performs as expected under various conditions.

5.2.4 Test Result

The graph illustrates the relationship between training epochs and the accuracy of a machine learning model used in crop recommendation, fertilizer recommendation, or plant disease prediction. Initially, the model has low accuracy (~0.2) due to random weight initialization, but as training progresses, accuracy improves rapidly, indicating effective learning from the data. Around the 100th epoch, accuracy plateaus, suggesting that the model has reached optimal performance, and further training may risk overfitting. While the model demonstrates good accuracy in predicting suitable crops and determining fertilizer needs, additional fine-tuning or larger datasets may be necessary for better performance, especially in accurately diagnosing rarer plant diseases. Overall, the results highlight the need for continuous evaluation and optimization of the models for real-world applicability.

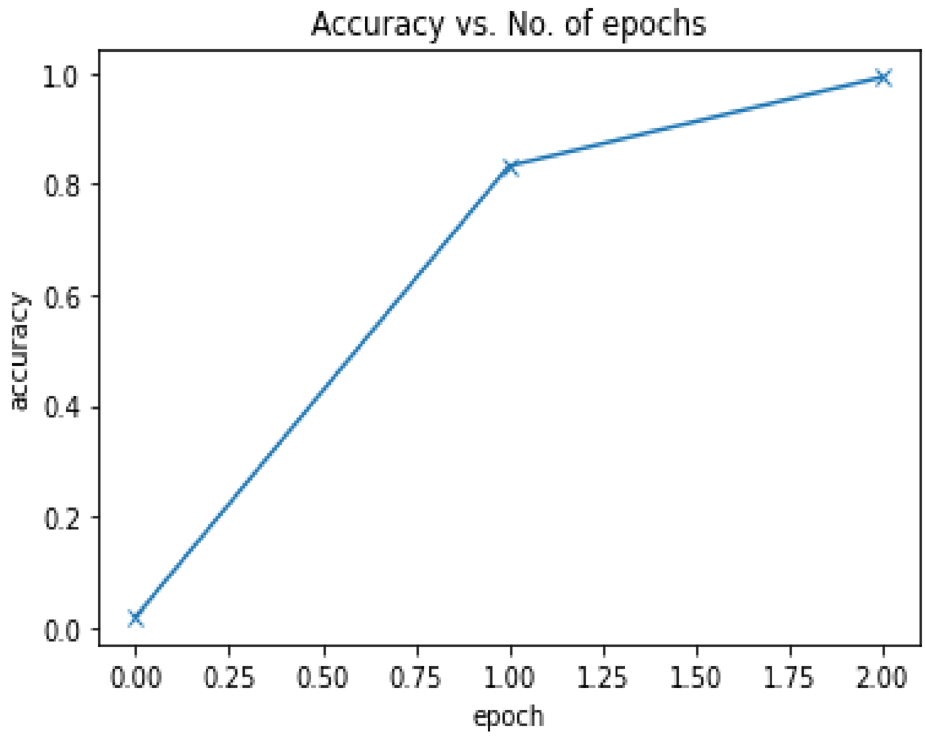

Figure 5.7: **Accuracy vs no. of epochs**

Key metrics for evaluating model performance include accuracy, precision, recall, and F1 Score. XGBoost outperforms all other models, achieving the highest accuracy, precision, recall, and F1 score, making it the best choice for the task. Random Forest also performs well, while Decision Tree, Support Vector Machine, Logistic Regression, and Naïve Bayes show reasonable accuracy but may have specific limitations. In cases of class imbalance, the F1 score is more informative than accuracy, and error analysis of misclassified examples can offer insights for improvement. Hyperparameter tuning could further enhance performance. Overall, XGBoost is the most promising model, though further analysis is necessary for a definitive choice.

XGBoost outperforms other models with the highest accuracy, precision, recall, and F1 score, making it the best choice. Random Forest also performs well, while other models show reasonable accuracy but have limitations. In cases of class imbalance, the F1 score is more informative than accuracy. Hyperparameter tuning and error analysis can further improve performance. XGBoost stands out but requires further analysis to confirm its suitability.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Decision Tree	85.0	82.0	80.0	81.0
Random Forest	90.5	88.0	87.5	87.7
Support Vector Machine	88.0	85.0	84.0	84.5
Logistic Regression	86.5	84.5	83.0	83.7
Naïve Bayes	83.5	81.0	79.0	80.0
XGBoost	92.0	90.0	89.0	89.5

Figure 5.8: **Performance Analysis**

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of the proposed Harvestify system is grounded in its ability to streamline data-driven agricultural practices through advanced technologies. First, the data processing pipeline is designed to handle large datasets from IoT sensors, user inputs, and external APIs, efficiently cleaning and preparing this data for analysis. Automated workflows for preprocessing ensure that noise removal and feature extraction occur quickly, enabling timely recommendations that are essential in dynamic farming environments. Next, the system leverages high-performance machine learning models such as Random Forest, XGBoost, and Support Vector Machines, optimized for specific tasks like crop recommendation, fertilizer suggestion, and plant disease detection. These algorithms are selected for their ability to handle complex data while minimizing computational load, resulting in fast and accurate predictions. By using parallel processing techniques and selecting features that most significantly impact outcomes, the system reduces the time and resources needed for training and evaluation, further enhancing its efficiency.Moreover, the user interface is designed to provide farmers with actionable insights quickly. By synthesizing the outputs from the machine learning models, the recommendation system generates personalized, easy-to-understand advice on crop selection, fertilizer application, and disease management, all in real-time. Finally, resource optimization is a core strength of the system, as it helps farmers reduce unnecessary use of fertilizers and pesticides, thus lowering costs and environmental impact. The system’s integration of real-time data with predictive analytics ensures that recommendations are based on the most current conditions, leading to better decision-making and higher efficiency in agricultural operations.

6.2 Comparison of Existing and Proposed System

The existing systems in agriculture are often limited in terms of technological integration and data utilization. They typically rely on basic data inputs, manual processes, and simple rule-based algorithms for decision-making, which restricts their ability to offer personalized, data-driven recommendations. These systems generally lack real-time data collection capabilities, relying on historical data that may not reflect current conditions, leading to less accurate and timely guidance for farmers. In contrast, the proposed Harvestify system introduces a more advanced approach by integrating modern technologies such as machine learning, deep learning, and IoT. Harvestify leverages real-time data collection from IoT sensors to monitor crucial factors like soil moisture, temperature, humidity, and nutrient levels. This data, combined with external sources like weather APIs, provides a comprehensive dataset for analysis. The system's use of sophisticated algorithms, such as Random Forest and XGBoost for crop recommendations and deep learning for plant disease detection, allows it to deliver far more accurate and tailored insights. Moreover, Harvestify's recommendation system is designed to be more dynamic and responsive compared to existing solutions. It uses real-time data and predictive models to generate personalized recommendations for crop selection, fertilizer application, and disease management. This ensures that farmers receive timely, context-specific guidance, which helps in optimizing yields and reducing unnecessary input costs. Overall, while existing systems provide a more static and generalized approach, the proposed Harvestify system significantly improves efficiency, accuracy, and adaptability through its integration of advanced technologies and real-time data analytics.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The Harvestify project offers a comprehensive solution for enhancing agricultural practices by leveraging the power of machine learning, deep learning, and IoT technologies. Through the integration of real-time data collection, advanced algorithms, and an intuitive recommendation system, Harvestify empowers farmers to make informed decisions about crop selection, fertilizer application, and plant disease management. By providing tailored recommendations based on local conditions and real-time data, Harvestify not only optimizes agricultural productivity but also promotes sustainability by minimizing the unnecessary use of resources. Overall, the system bridges the gap between traditional farming methods and modern technological advancements, delivering a user-friendly platform that improves decision-making, increases crop yields, and reduces input costs. Harvestify stands as a valuable tool for modern farming, offering personalized insights that are tailored to local conditions. In essence, Harvestify represents a shift toward precision agriculture, where decisions are informed by data, ultimately leading to increased yields, reduced costs, and a more sustainable agricultural ecosystem.

7.2 Future Enhancements

The future of Harvestify holds exciting possibilities for further advancements and features. One of the primary areas for growth is the expansion of the crop and disease database, which would allow the system to offer recommendations for a wider variety of crops and diseases, accommodating more regions and crop types. Additionally, integration with advanced imaging technologies, such as drones and satellite data, can enable more detailed monitoring of crop health, pest infestations, and field conditions, improving the precision of recommendations. Another potential enhancement is the use of predictive analytics for market trends and yield forecasting, allowing farmers to not only manage current crops but also plan for future planting seasons based on predicted market demands and environmental changes. Incorporating blockchain technology for transparent supply chain management and traceability could also help farmers ensure the quality and origin of their produce, increasing consumer trust. In terms of accessibility, developing a mobile application would allow farmers to interact with the system in real-time from remote locations, improving the ease of use. Offline functionality could be implemented to ensure that farmers in regions with limited internet connectivity can still use the system for data collection and receive recommendations when online again. Additionally, multilingual support would enhance usability for farmers in different parts of the world, allowing Harvestify to be a truly global platform. Finally, incorporating user feedback loops for continuous learning and model refinement will ensure that the system adapts over time, providing more relevant and accurate predictions. By focusing on these future enhancements, Harvestify can continue to evolve and better meet the needs of farmers globally, driving innovation in sustainable agriculture.

CHAPTER 8

SOURCE CODE

```
# Importing essential libraries and modules

from flask import Flask, render_template, request, Markup
import numpy as np
import pandas as pd
from utils.disease import disease_dic
from utils.fertilizer import fertilizer_dic
import requests
import config
import pickle
import io
import torch
from torchvision import transforms
from PIL import Image
from utils.model import ResNet9
#
    ===============================================================
    ========================================

# -------------------------LOADING THE TRAINED MODELS -----------------------------
    -----------------

# Loading plant disease classification model

disease_classes = ['Apple___Apple_scab',
            'Apple___Black_rot',
            'Apple___Cedar_apple_rust',
            'Apple___healthy',
            'Blueberry___healthy',
            'Cherry_(including_sour)___Powdery_mildew',
            'Cherry_(including_sour)___healthy',
            'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
            'Corn_(maize)___Common_rust_',
            'Corn_(maize)___Northern_Leaf_Blight',
            'Corn_(maize)___healthy',
            'Grape___Black_rot',
            'Grape___Esca_(Black_Measles)',
            'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
            'Grape___healthy',
```

```
                   'Orange___Haunglongbing_(Citrus_greening)',
                   'Peach___Bacterial_spot',
                   'Peach___healthy',
                   'Pepper,_bell___Bacterial_spot',
                   'Pepper,_bell___healthy',
                   'Potato___Early_blight',
                   'Potato___Late_blight',
                   'Potato___healthy',
                   'Raspberry___healthy',
                   'Soybean___healthy',
                   'Squash___Powdery_mildew',
                   'Strawberry___Leaf_scorch',
                   'Strawberry___healthy',
                   'Tomato___Bacterial_spot',
                   'Tomato___Early_blight',
                   'Tomato___Late_blight',
                   'Tomato___Leaf_Mold',
                   'Tomato___Septoria_leaf_spot',
                   'Tomato___Spider_mites Two-spotted_spider_mite',
                   'Tomato___Target_Spot',
                   'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
                   'Tomato___Tomato_mosaic_virus',
                   'Tomato___healthy']

disease_model_path = 'models/plant_disease_model.pth'
disease_model = ResNet9(3, len(disease_classes))
disease_model.load_state_dict(torch.load(
    disease_model_path, map_location=torch.device('cpu')))
disease_model.eval()

# Loading crop recommendation model

crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
    open(crop_recommendation_model_path, 'rb'))


           # Custom functions for calculations

def weather_fetch(city_name):
    """
    Fetch and returns the temperature and humidity of a city
    :params: city_name
    :return: temperature, humidity
    """
```

```
    api_key = config.weather_api_key
    base_url = "http://api.openweathermap.org/data/2.5/weather?"

    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()

    if x["cod"] != "404":
        y = x["main"]

        temperature = round((y["temp"] - 273.15), 2)
        humidity = y["humidity"]
        return temperature, humidity
    else:
        return None

def predict_image(img, model=disease_model):
    """
    Transforms image to tensor and predicts disease label
    :params: image
    :return: prediction (string)
    """
    transform = transforms.Compose([
        transforms.Resize(256),
        transforms.ToTensor(),
    ])
    image = Image.open(io.BytesIO(img))
    img_t = transform(image)
    img_u = torch.unsqueeze(img_t, 0)

    # Get predictions from model
    yb = model(img_u)
    # Pick index with highest probability
    _, preds = torch.max(yb, dim=1)
    prediction = disease_classes[preds[0].item()]
    # Retrieve the class label
    return prediction



# ------------------------------------ FLASK APP -------------------------------------------------

app = Flask(__name__)

# render home page
```

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

# render crop recommendation form page

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

# render fertilizer recommendation form page

@ app.route('/fertilizer')
def fertilizer_recommendation():
    title = 'Harvestify - Fertilizer Suggestion'

    return render_template('fertilizer.html', title=title)

# render disease prediction input page
#
        ================================================================
        ===========================================

# RENDER PREDICTION PAGES

# render crop recommendation result page

@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'Harvestify - Crop Recommendation'

    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        if weather_fetch(city) != None:
```

```
            temperature, humidity = weather_fetch(city)
            data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
            my_prediction = crop_recommendation_model.predict(data)
            final_prediction = my_prediction[0]

            return render_template('crop-result.html', prediction=final_prediction,
           title=title)

        else:

            return render_template('try_again.html', title=title)

# render fertilizer recommendation result page

@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'Harvestify - Fertilizer Suggestion'

    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['pottasium'])
    # ph = float(request.form['ph'])

    df = pd.read_csv('Data/fertilizer.csv')

    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
    elif max_value == "P":
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
```

```
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"

    response = Markup(str(fertilizer_dic[key]))

    return render_template('fertilizer-result.html', recommendation=response, title=title)

# render disease prediction result page

@app.route('/disease-predict', methods=['GET', 'POST'])
def disease_prediction():
    title = 'Harvestify - Disease Detection'

    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files.get('file')
        if not file:
            return render_template('disease.html', title=title)
        try:
            img = file.read()

            prediction = predict_image(img)

            prediction = Markup(str(disease_dic[prediction]))
            return render_template('disease-result.html', prediction=prediction, title=title)
        except:
            pass
    return render_template('disease.html', title=title)

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

REFERENCES

[1] Sachin Adulkar, Vivek Pawar, Aniket Choudhari, Shubham Kothekar, Shruti Agrawal", Crop Disease Detection and Fertilizer Suggestion",2021 International Journal of Scientific Research in Computer Science, Engineering and Information Technology.

[2] Miss. Madhura Patil, Miss. Sakshi Surve, Mr. Akshay Tembe," Harvesting System Using Machine",2018 International Journal of Computer Applications.

[3] Shima Ramesh Maniyath, Niveditha M, Pooja R, Prasad Bhat N, Shashank N, Mr. Ramachandra Hebbar, Mr. P V Vinod," Plant Disease Detection Using Machine Learning",2020 International Journal of Scientific Research in Computer Science, Engineering and Information Technology.

[4] Doshi, Zeel, Subhash Nadkarni, Rashi Agrawal, and Neepa Shah. "AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms",2019 International Journal of Novel Research and Development.

[5] Suresh, G., A. Senthil Kumar, S. Lekashri, and R. Manikandan. "Efficient Crop Yield Recommendation System Using Machine Learning For Digital Farming",2021 International Journal of Advanced Research in Computer and Communication Engineering.

[6] Dighe, Deepti, Harshada Joshi, Aishwarya Katkar, Sneha Patil, and Shrikant Kokate. "Survey of Crop Recommendation Systems",2022 Institute of Electrical and Electronics Engineer.