TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KALANKI, KATHMANDU

[CT755]

A Major Project Report On "CropAI: Intelligent Crop Yield and Fertilizer Recommendation System"

SUBMITTED BY:

Subash Limbu (ACE076BCT086)

Sushant Tripathee (ACE076BCT090)

Samyak Deshar (ACE076BCT074)

Samir Ghimire (ACE076BCT072)

SUPERVISED BY:

Associate Prof. Anand Kumar Sah

A Major Project Final report submitted to the department of Electronics and Computer Engineering in the partial fulfillment of the requirements for degree of Bachelor of Engineering in Computer Engineering

Kathmandu, Nepal

Feb 23, 2024

CropAI: Intelligent Crop Yield and Fertilizer Recommendation System

Submitted by:

Subash Limbu (ACE076BCT086)

Sushant Tripathee (ACE076BCT090)

Samyak Deshar (ACE076BCT074)

Samir Ghimire (ACE076BCT072)

Supervised by:

Anand Kumar Sah

(Associate Professor)

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

"DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING"

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

Balkhu, Kathmandu

Feb 23, 2023

LETTER OF APPROVAL

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled "**CropAI: Intelligent Crop Yield and Fertilizer Recommendation System**" submitted by:

Subash Limbu (ACE076BCT086)

Sushant Tripathee (ACE076BCT090)

Samyak Deshar (ACE076BCT074)

Samir Ghimire (ACE076BCT072)

In the partial fulfillment of the requirements for the degree of Bachelor's Degree in Computer Engineering.

………………………..

Project Supervisor

Anand Kumar Sah **(Associate Professor)**

Department of Electronics and Computer Engineering

………………………..

Head of Department

Senior Asst. Prof. Prem Chandra Roy

Department of Electronics and Computer Engineering

Feb 23, 2023

COPYRIGHT

Head of Department
Department of Electronics and Computer Engineering
Advanced College of Engineering and Management
Balkhu, Kathmandu
Nepal

ACKNOWLEDGEMENT

We take this opportunity to express our deepest and sincere gratitude to our Project Supervisor, Associate Professor Er. **Anand Kumar Sah** for his/her insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his/her constant encouragement and advice throughout our Bachelor's programme.

We express our deep gratitude to **Er. Prem Chandra Roy**, Head of Department of Electronics and Computer Engineering, **Er. Bikash Acharya, Er. Dhiraj Pyakurel**, Deputy Head, Department Of Electronics and Computer Engineering, **Er. Laxmi Prasad Bhatt**, Academic Project Coordinator, Department Of Electronics and Computer Engineering for their regular support, co-operation, and coordination.

The in-time facilities provided by the department throughout the Bachelors program are also equally acknowledgeable.

We would like to convey our thanks to the teaching and non-teaching staff of the Department of Electronics & Communication and Computer Engineering, ACEM for their invaluable help and support throughout the period of Bachelor's Degree. We are also grateful to all our classmates for their help, encouragement and invaluable suggestions.

Finally, yet more importantly, I would like to express our deep appreciation to my grandparents, parents, siblings for their perpetual support and encouragement throughout the Bachelor's degree period.

Subash Limbu (ACE076BCT086)
Sushant Tripathee (ACE076BCT090)
Samyak Deshar (ACE076BCT074)
Samir Ghimire (ACE076BCT072)

ABSTRACT

This report presents a comprehensive system for recommending the best crops and appropriate fertilizers to farmers based on soil properties and weather conditions. The system utilizes sensors, WiFi module, GPS devices and an Arduino to extract soil properties such as pH, nitrogen, phosphorus, and potassium, as well as weather conditions including climate and rainfall. Machine learning algorithms, including Random Forest, Support Vector Machine(SVM), Decision Tree and Multilayer Perceptron, are employed to analyze the collected data and provide accurate crop recommendations. Data is collected from various sources, including the aforementioned sensors and modules, as well as historical crop yield data from agricultural databases. The collected data is preprocessed and cleaned, ensuring its quality and reliability. Feature engineering techniques are applied to extract relevant features from the data, allowing for more effective analysis. The chosen machine learning algorithms, namely Random Forest, Support Vector Machine, Decision Tree, and Multilayer Perceptron, are trained on the training data, optimizing their hyperparameters to achieve the best performance.

Evaluation of the trained models is conducted using the test set. The most accurate and reliable model is selected as the final recommendation system. The system is designed to provide farmers with an intuitive user interface that takes input from the sensors and modules, feeds it into the trained model, and displays the recommended crops. Throughout the project, a strong emphasis is placed on continuous monitoring and feedback gathering from farmers. By leveraging the capabilities of sensors, WiFi module, Arduino, and machine learning algorithms, this project aims to empower farmers with data-driven insights, facilitating informed decision-making in crop selection and optimizing agricultural practices.

Keywords: *Arduino uno, Crop recommendation, Decision Tree, deep learning, machine learning, Multilayer Perceptron (MLP), NPK soil sensor, Ph sensor, Arduino, Random Forest, sensors, Support Vector Machine (SVM), soil properties, weather conditions, Wi-Fi module.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DFD	Data flow Diagram
GPU	Graphical Processing Unit
GPS	Global Positioning System
GUI	Graphical User Interface
HTML	HyperText Markup Language
IDE	Integrated Development Environment
ML	Machine Learning
MLP	Multilayer Perceptron
PH	Power of hydrogen
RAM	Random Access Memory
RF	Random Forest
SVM	Support Vector Machine

List Of Units And Conversions

pH	Power of hydrogen
mg/kg	Milligram/Kilogram
mm/year	Millimeter/Year
°C	Degree Celsius
m	Meter

CHAPTER 1: INTRODUCTION

The agriculture sector plays a critical role in sustaining global food production and meeting the increasing demands of a growing population. However, farmers often face challenges in determining the best crop to cultivate based on varying soil properties and weather conditions. Traditional methods of crop selection rely on experience and intuition, which may not always result in optimal outcomes. To address this issue, this report presents a comprehensive system that utilizes sensors, Wifi module, and an Arduino to extract soil properties and weather conditions. Machine learning algorithms, including Random Forest, Decision Tree, Support Vector Machine and Multi-layer Perceptron, are then employed to analyze the collected data and provide crop recommendations based on historical patterns.

1.1: Background

Nepal's agricultural landscape is diverse, with varying soil types and climatic conditions across different regions. Factors such as soil pH, nutrient levels, and moisture content have a direct impact on crop growth and productivity. Additionally, weather conditions, including rainfall patterns, temperature, and humidity, significantly influence crop performance and disease prevalence. To address these challenges and support farmers in Nepal, there is a growing need for advanced technologies and data-driven solutions. Integrating sensors, Wifi modules, and the power of Arduino can enable the collection of real-time data on soil properties and weather conditions. These technologies provide an opportunity to gather precise and localized information, allowing farmers to make well-informed decisions tailored to their specific agricultural contexts [1]. Machine learning algorithms, including Random Forest, Decision Tree, Support Vector Machine and Multi-layer Perceptron, offer a promising approach for analyzing the collected data [2]. By leveraging historical data on crop yields and combining it with real-time information, these algorithms can identify patterns and correlations between soil properties, weather conditions, and crop performance [3]. This enables the development of a crop recommendation system that suggests the most suitable crops based on the prevailing conditions.

1.2: Motivation

The motivation behind this project is to empower farmers in agricultural countries like Nepal with data-driven insights for informed decision-making. By integrating sensors, Wifi modules, and machine learning algorithms, this project aims to provide farmers with real-time information on soil properties and weather conditions. The ultimate goal is to enhance crop selection processes, optimize agricultural practices, and maximize yields, contributing to the overall growth and sustainability of the agricultural sector in Nepal.

1.3: Problem Statement

The agricultural sector in Nepal faces significant challenges due to the absence of a comprehensive crop and fertilizer recommendation system. Farmers struggle with making informed decisions about crop selection and fertilizer application, resulting in reduced agricultural productivity, inefficient resource utilization, and environmental degradation. The lack of access to accurate and personalized agricultural information, combined with the diverse soil variability across different regions of Nepal, further compounds these challenges.

The objective is to overcome these challenges and develop a comprehensive solution that empowers farmers in agricultural countries to optimize their agricultural practices, improve crop productivity, and promote sustainable farming. By addressing the problem of inaccurate yield estimation and suboptimal fertilizer application, the project aims to enhance agricultural efficiency, contribute to food security, and mitigate the environmental impact of farming practices.

1.4: Objective

The objective of this project is to develop a crop recommendation system that utilizes sensors, Wifi module, Arduino and machine learning algorithms to provide farmers in agricultural countries like Nepal with accurate and timely recommendations for crop selection based on soil properties and weather conditions. The system aims to empower farmers, optimize agricultural practices, and enhance overall productivity and sustainability in the agricultural sector.

1.4.1: General Objective

The objective of our project is to develop a crop recommendation system for farmers in agricultural countries, particularly Nepal, using sensors, Wifi module, Arduino, and machine learning algorithms to provide accurate and timely crop selection recommendations based on soil properties and weather conditions.

1.4.2: Specific Objective

Following are the specific objectives collectively aim to achieve the general objective of enhancing agricultural productivity and sustainability through an advanced crop and fertilizer recommendation system.

- Integrate soil and weather sensors for real-time data collection.
- Implement Wifi modules and Arduino microcontrollers for seamless data transmission and processing.
- Design a robust database for efficient storage and retrieval of sensor data.
- Develop and optimize machine learning algorithms for crop and fertilizer recommendations.
- Create an accessible web or mobile interface for farmers to interact with the system.
- Conduct field trials to validate system accuracy under various conditions.
- Customize the system to align with specific agricultural practices in Nepal.

1.5: Significance of the study

The study of this project holds an immense significance, especially in an agricultural country. It contributes in the following aspects.

- Improved Crop Selection: The study aims to provide farmers with data-driven recommendations for crop selection based on soil properties and weather conditions. This can lead to more informed decision-making and increase the likelihood of selecting crops that are best suited for specific conditions, leading to improved yields and profitability.
- Resource Optimization: By utilizing the recommended crop selection, farmers can optimize the use of resources such as fertilizers, water, and pesticides. This can help reduce resource wastage and minimize environmental impact while maintaining or improving crop productivity.
- Increased Productivity: The study aims to enhance overall productivity in the agricultural sector by guiding farmers towards the most suitable crops for their specific conditions. This can contribute to increased yields, improved food security, and economic growth in agricultural communities.
- Sustainability and Environmental Impact: By promoting optimal crop selection based on soil properties and weather conditions, the study can contribute to sustainable agricultural practices. This includes reducing the use of agrochemicals, minimizing soil erosion, and conserving water resources, leading to a reduced environmental impact.
- Technology Adoption: The study leverages advanced technologies such as sensors, Wifi modules, and machine learning algorithms, showcasing the potential of integrating technology into agriculture. This can encourage the adoption of modern agricultural practices and pave the way for future technological advancements in the agricultural sector.
- Empowering Farmers: By providing farmers with access to real-time data and intelligent recommendations, the study aims to empower them with valuable information for decision-making. This can enhance their confidence, improve their livelihoods, and enable them to make more informed choices regarding crop selection and agricultural management.
- Knowledge Expansion: The study explores the application of machine learning algorithms in the context of crop recommendation, showcasing the potential of these approaches in improving agricultural outcomes.

CHAPTER 2: LITERATURE REVIEW

In agriculture, Machine Learning is considered a novel field, as a variety of work has been done with the help of machine learning in the field of agriculture. There are different philosophies made and evaluated by the researchers all through the world in the field of agriculture and related sciences.

- Anguraj. K, Thiyaneswaran. B, Megashree. G, Preetha Shri. J. G, Navya. S, and Jayanthi. J. [5] utilized machine learning for crop suggestions based on soil analysis. They utilized an IoT system to gather live data, which was employed to train a model for predictive analysis. The model's outcomes substantially assist in selecting appropriate crops for specific field areas. This approach achieved an accuracy level of roughly 75%.

- R. Pallavi Reddy, B. Vinitha, K. Rishitha, and K. Pranavi [6] developed crop monitoring and recommendation systems using machine learning and IoT. Sensors collect data sent to NodeMCU, an Android app used for model training and predictions. The system's algorithm, along with components like sensors and Arduino, is employed to build the modules. The resultant system provides relevant guidance and is easy to use.

- Aakunuri Manjula and Dr. G. Narsimha developed a crop recommendation and yield forecast for agriculture using data mining methods. Steps in this methodology include collecting data, selecting features, utilizing classification techniques, and suggesting crops through ensembling algorithms. The accuracy and precision of the system were limited due to the utilization of data mining techniques [7].

- Mahendra Choudhary, Rohit Sartandel, Anish Arun, and Leena employed machine learning in precision agriculture to create a crop recommendation system and classify plant diseases. The system's methodology involves categorizing agricultural diseases and encompasses stages such as data collection, exploration, division, and application. They divided the main dataset into training, validation, and testing sets using a ratio of 60:20:20. They utilized the Random Forest Classifier to predict crops [8].

- L. C. Gavade, [9] explains using fiber optics, humidity, temperature and sunlight sensors to estimate the NPK values of soil and other parameters and pass it to the microcontroller to remotely control the work in the agriculture field such as fertilization, pumping of water depending on the sensor readings. This paper gives the basic structure for smart agriculture.

CHAPTER 3: REQUIREMENT ANALYSIS

3.1: Software Implementations

- Arduino IDE : The Arduino Integrated Development Environment (IDE) is an open-source software platform used for programming Arduino microcontrollers. It provides a user-friendly interface for writing, compiling, and uploading code to Arduino boards.

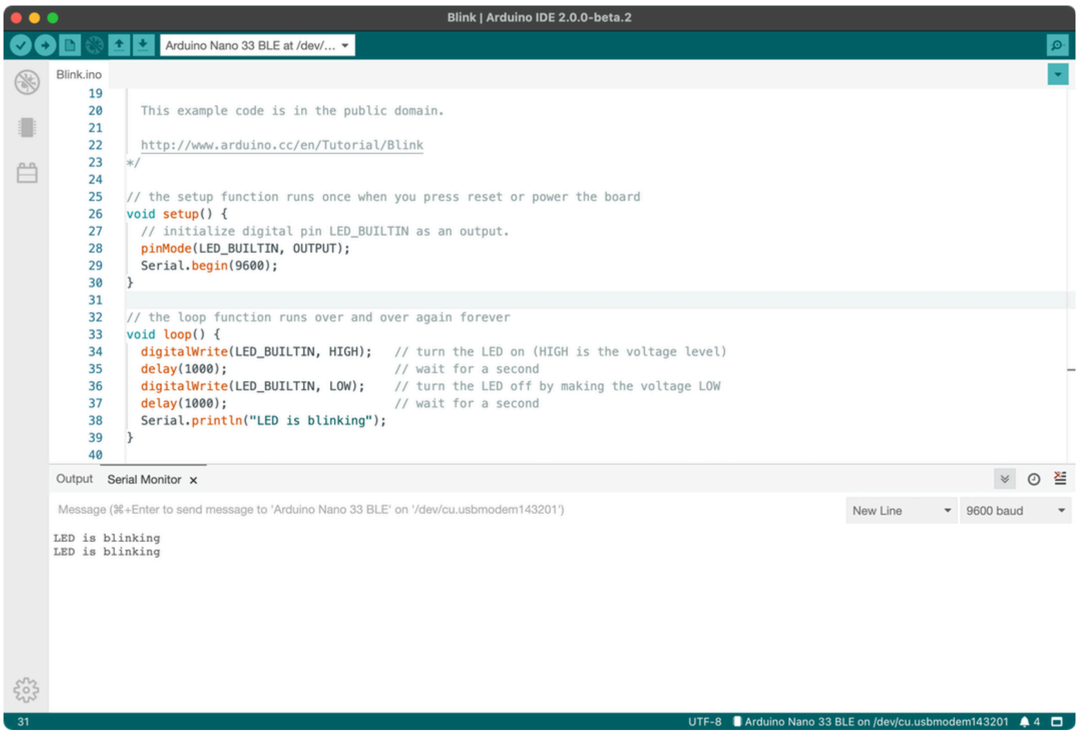

Figure 3.1.1: Arduino IDE

[Source:https://www.datocms-assets.com/76605/1659348717-software-arduino-pro-ide.png

Accessed February 16,2023]

- Visual Studio Code : Visual Studio Code (VS Code) is a popular and versatile source code editor developed by Microsoft. It offers a wide range of features, including syntax highlighting, code completion, and debugging tools, making it suitable for various programming languages and development tasks. With its customizable interface and extensive library of extensions, developers can tailor VS Code to suit their specific needs and workflows.

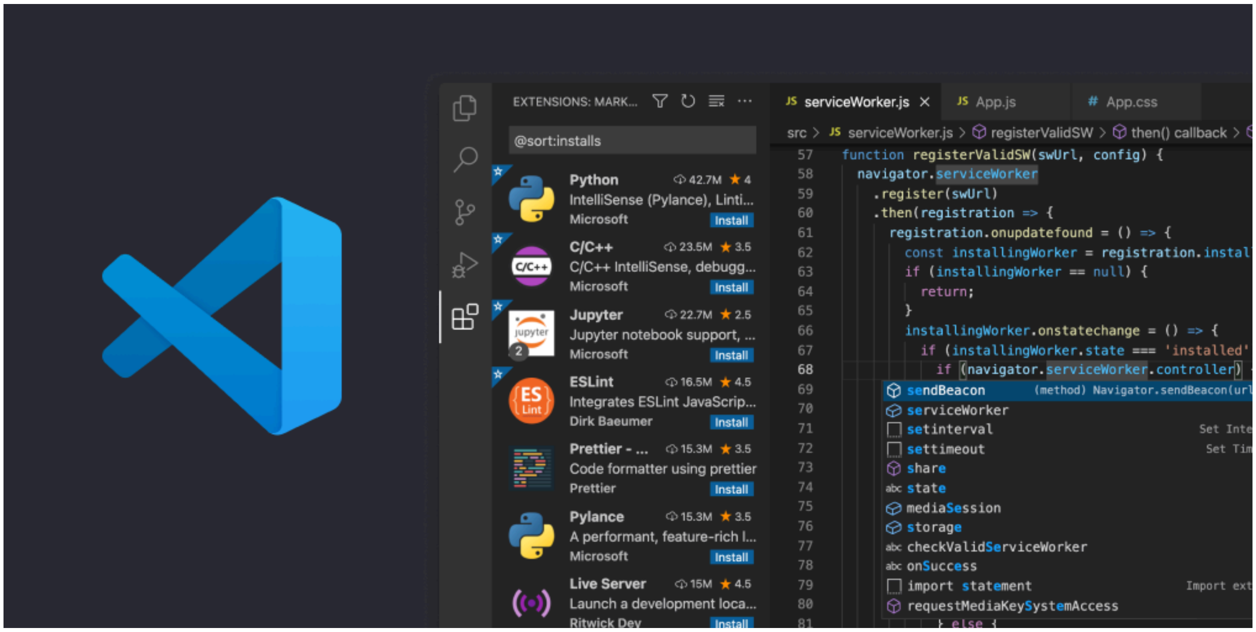

Figure 3.1.2: VS Code

[Source: https://code.visualstudio.com/opengraphimg/opengraph-home.png Accessed February 16,2024]

- Python: Flask is a web framework that uses Python, so you need to have Python installed on your system.
- Flask: Flask is a lightweight web framework for Python. It’s a Python module that lets you develop web applications easily.
- Machine Learning Libraries: Since you're building a prediction model, you will need machine learning libraries in Python. Some popular libraries for machine learning are scikit-learn, TensorFlow, or PyTorch.
- Data Processing Libraries: You need libraries for data manipulation and preprocessing, such as pandas and NumPy.
- HTML/CSS/JavaScript: For building the user interface, you will need HTML, CSS, and possibly JavaScript. Flask integrates well with these front-end technologies.

3.2: Hardware Implementations

Here are general hardware requirements that we need to run this model:

- Microcontroller(Arduino UNO): A microcontroller or a development board is needed to interface with the soil sensor tester and process the sensor data. Popular options include Arduino boards (e.g., Arduino Uno, Arduino Mega) or Raspberry Pi. The Arduino Uno REV3 is a popular microcontroller board developed by the Arduino Foundation. It is widely used for prototyping and creating various electronics projects due to its simplicity, versatility, and affordability. The Arduino Uno REV3 boards are widely used in various projects and applications, ranging from education and hobbyist projects to industrial applications and DIY electronics.

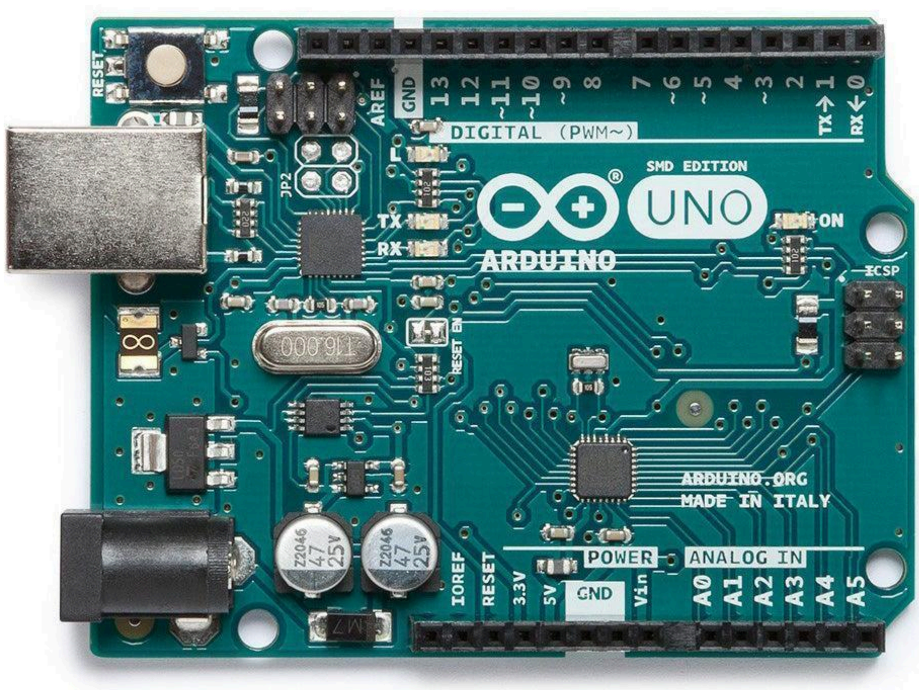

Figure 3.2.1: Arduino Uno REV3

[Source:https://cdn.shopify.com/s/files/1/0174/1800/products/a000073_front_1_93d36068-b342-4564-8cca-ae2feef26891_1024x1024.jpg?v=1530371789 Accessed August 9,2023]

- Bread Board : A breadboard is a reusable prototyping tool used in electronics to create temporary circuits without soldering. It consists of a plastic board with metal clips underneath, arranged in rows and columns. Components like resistors, LEDs, and integrated circuits are inserted into the board's holes, making connections through the clips. Power rails along the edges provide voltage and ground connections. Breadboards are popular for quickly testing and iterating circuit designs before permanent assembly, making them essential for hobbyists, students, and professionals alike.

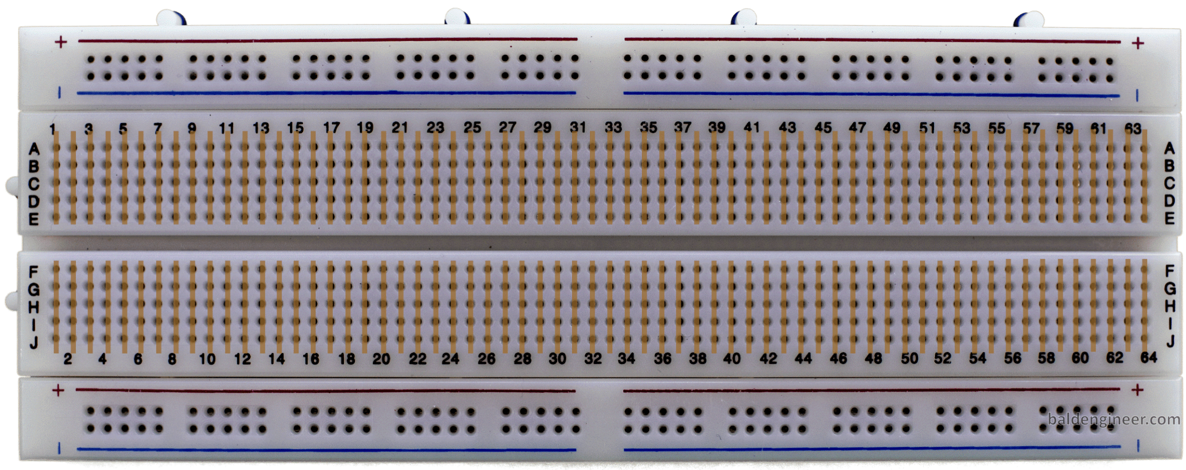

Figure 3.3.5: Bread Board

[Source:https://www.baldengineer.com/wp-content/uploads/2013/12/breadboard_component-columns-highlighted.png
Accessed august 9,2023]

- Jumper Cable: A jumper wire, also known as a jumper or DuPont wire, is an electrical wire with a connector or pin at each end. These wires are commonly used to interconnect components on a breadboard or other prototype circuits without the need for soldering. Types of jumper cable: Male-to-male, Male-to-female, Female-to-female. Male-to-male: These jumpers have pins at both ends and can plug into various components. Male-to-female: Male pins on one end and female connectors on the other. Female-to-female: Both ends have female connectors.

Figure 3.2.6: Jumper cable (male-female jumper cable)

[Source:https://i5.walmartimages.com/asr/a12e7383-52da-4b4f-9679-e71f99b36f31_1.443ffff689700267fcd0a532205828c0.jpeg

Accessed august 9,2023]

- Ph meter/Sensor : A pH sensor is a device used to measure the acidity or alkalinity of a solution. pH is a measure of the concentration of hydrogen ions in a solution, and it's expressed on a scale from 0 to 14, where 0 is highly acidic, 7 is neutral, and 14 is highly alkaline (basic).

Figure 3.2.2: Ph Sensor

[*https://ae01.alicdn.com/kf/HTB1k0XamDmWBKNjSZFBq6xxUFXaU.jpg*

Accessed December 29 ,2023]

- NPK Soil Sensor : These sensors are used to measure the levels of nitrogen(N), phosphorus(P), and potassium(K) in the soil. There are various NPK sensors available in the market that can be connected to a microcontroller or a data acquisition system. The soil NPK sensor is suitable for detecting the content of nitrogen, phosphorus, and potassium in the soil, and judging the fertility of the soil by detecting the content of N, P, and K in the soil. The stainless steel probe of the soil NPK sensor can be buried in the soil for a long time and is resistant to long term electrolysis, salt, and alkali corrosion.

Figure 3.2.3: NPK Soil Sensor

[Source: https://www.renkeer.com/wp-content/uploads/2021/06/soil-npk-sensor-2.jpg Accessed August 9,2023]

- Humidity and Temperature Sensor (DHT22): The DHT22 is a type of humidity and temperature sensor that provides accurate and reliable measurements of both humidity and temperature. It features a capacitive humidity sensor and a thermistor for temperature sensing, all integrated into a single module. The DHT22 sensor communicates with microcontrollers using a digital protocol, typically a single-wire interface. This sensor is known for its high accuracy and stability over time, making it suitable for various applications, including weather stations, environmental monitoring systems

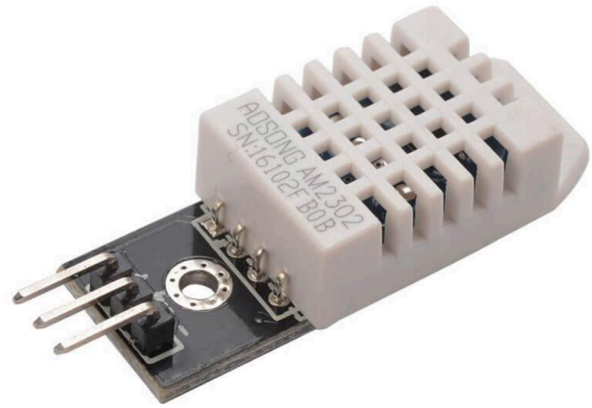

Figure 3.2.4 : DHT22 (Humidity and Temperature Sensor)

[Source:https://www.flyrobo.in/image/cache/catalog/dht22-digital-temperature-and-humidity-sensor-module/dht22-digital-temperature-and-humidity-sensor-module4-550x550.jpg

Accessed December 28 ,2023]

- GPS Module: A GPS module is a device that allows your device or circuit to receive GPS data. GPS stands for Global Positioning System, which is a satellite-based radio navigation system that determines precise location, velocity, and time. A GPS module can be used for tracking, navigation, or timing applications

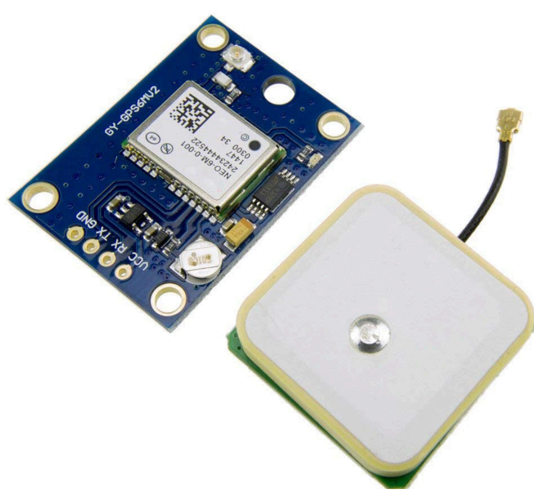

Figure 3.2.7: GPS Module (ublox neo-6m)

[Source:https://digitalelectronics.lk/wp-content/uploads/2020/03/NEO-6M-GPS-Module-with-Active-Antenna1.jpg

Accessed august 9,2023]

- Wifi module: A connectivity module to transmit the sensor data wirelessly to a central server or a web application. It's widely used in IoT (Internet of Things) projects and offers Wi-Fi connectivity for devices like microcontrollers.

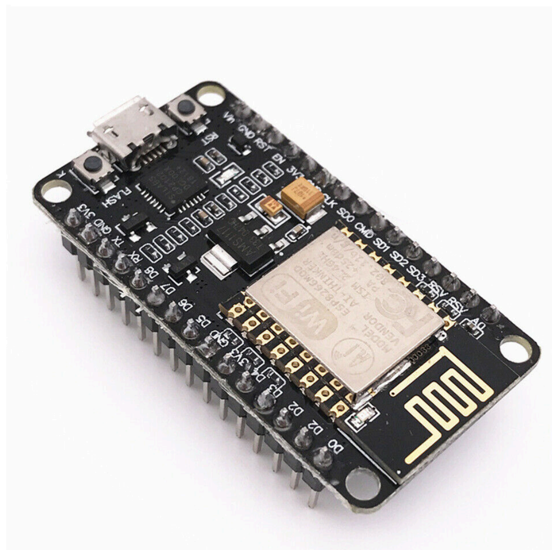

Figure 3.2.8: Wifi Module (esp8266)

[Source:https://www.okystar.com/wp-content/uploads/2019/07/OKY22514.jpg

Accessed August 5,2023]

- Power Supply: It provides a stable power supply to the microcontroller and sensors. This can be achieved using batteries, power adapters, or solar panels, depending on the deployment scenario.
- GPU (optional): While not strictly necessary, having a dedicated Graphics Processing Unit (GPU) can significantly speed up the training and prediction process, especially for deep learning models. GPUs are highly parallel processors that excel at matrix operations, which are prevalent in ML algorithms. If your model is computationally intensive or uses deep learning algorithms, consider using a GPU.

- Storage: You need storage space to store your ML model, dataset, and any other related files. The size of storage depends on the size of the model and dataset you are working with. Ensure you have enough disk space available for these files.
- Processor (CPU): A multicore processor or server-grade processor is recommended to handle the computational load. ML predictions can be computationally intensive, especially when dealing with large datasets or complex models.
- Memory (RAM): Sufficient RAM is essential to store and manipulate the data required for predictions. The amount of memory required depends on the size of your dataset and the complexity of your ML model. Generally, 8 GB or more of RAM is recommended, but this can vary depending on your specific use case.

3.3: Non-functional requirements

Some non-functional requirements of the proposed system are:-

- Performance: The system should provide timely recommendations, with minimal delay, to ensure farmers can make informed decisions in a timely manner. It should be able to handle a significant number of users and process large datasets efficiently to avoid performance bottlenecks.
- Scalability: It should be scalable to accommodate increasing data volumes and user demands as the user base grows. The system should be designed to handle varying scales of usage, from individual farmers to large agricultural operations, without compromising performance or accuracy.
- Reliability: The system should be reliable, consistently delivering accurate recommendations without unexpected failures or crashes.
- Security: The system should prioritize the security of user data, implementing robust security measures to protect against unauthorized access, data breaches, and data manipulation.
- Usability: The system should have an intuitive and user-friendly interface that is easy to navigate and understand, catering to users with varying levels of technical expertise. Clear instructions and guidance should be provided to assist users in inputting data and interpreting recommendations.
- Maintainability: The system should be designed and developed using modular and well-documented code, making it easier to maintain and update in the future. It should allow for easy incorporation of new climate and soil data sources or updates to existing algorithms and models.
- Compatibility: The system should be compatible with various devices, operating systems, and web browsers, ensuring accessibility across different platforms. It should also be capable of integrating with existing agricultural technologies or software solutions that farmers may already be using.
- Documentation: Provide comprehensive and up-to-date documentation, including installation instructions, system architecture, API documentation, and user guides.

3.4: Functional requirements

Here are some functional requirements that are consider:

- Data Collection and Integration: The system should gather and integrate climate data from reliable sources, including temperature, rainfall and humidity. Soil data, such as pH level, amount of nitrogen, phosphorus and potassium, should be collected and incorporated into the system.
- Data Analysis and Modeling: The system should employ algorithms and models to analyze the collected climate and soil data. Statistical techniques, machine learning, or artificial intelligence approaches can be used to derive insights and correlations between the data variables.
- Crop and Fertilizer Recommendation: Based on the analyzed data, the system should recommend suitable crop options that are compatible with the climate and soil conditions of the user's location. The system should suggest appropriate fertilizer types, quantities, and application methods based on the crop recommendations and the specific soil nutrient deficiencies identified.
- User Interface: The system should have an intuitive and user-friendly interface for easy interaction and input of data. Clear and understandable recommendations should be provided to the users, along with any relevant supporting information or explanations.

CHAPTER 4: METHODOLOGY

4.1: Data Collection And Dataset Creation

4.1.1: Data Sources Identification:

- Literature Review: Begin by conducting an extensive literature review to identify reliable sources of agricultural data. This includes research papers, government reports, INGO publications, and online repositories.
- Government Agencies and Institutions: Gather information from official government agricultural departments like NSSRC[a], research institutions, and agricultural extension services. These entities often provide comprehensive and validated data.
- Online Platforms: Explore online platforms such as Kaggle[b], data portals, and agricultural databases. These platforms might offer datasets related to crop growth conditions, soil nutrients, and climate parameters.
- Collaboration with Experts: Establish collaborations with agricultural experts, researchers, and agronomists. Expert insights can guide data collection and validation, ensuring the accuracy and relevance of the dataset.

4.1.2: Data Collection and Compilation:

- Crop Selection: Determine a diverse set of crops for inclusion in the dataset. Consider crops that are relevant to your region, have economic significance, and showcase a wide range of growth conditions.
- Attribute Collection: Collect essential attributes for each crop, including growth requirements (altitude, temperature, humidity, rainfall), soil nutrient content (Nitrogen, Phosphorus, Potassium), and potential nutrient deficiencies.
- Data Validation: Cross-reference data points across multiple sources to validate accuracy. Consult with experts to verify growth conditions and nutrient requirements specific to each crop.
- Normalization: Standardize units, scales, and measurement formats across different data sources to ensure uniformity. Convert data to a common format for seamless integration.

4.2: Data Collection Using Sensors, GPS, Wifi Module, and Arduino

4.2.1: Soil Input from NPK sensor and Ph sensor

- Sensor Integration: Install NPK sensors in the soil at specified locations. These sensors measure pH, Nitrogen, Phosphorus, and Potassium levels .
- DHT22(Humidity and soil sensor): This sensor measures the temperature and humidity of the particular location.
- Data Acquisition: Connect the NPK sensor, Ph sensor and Dht22 sensor are connected to Arduino. Read sensor data periodically using the Arduino.
- Data Processing: Process sensor data to obtain NPK, temperature, pH and humidity values. Calibrate and normalize data for accurate input.
- Integration: Integrate processed sensor data with crop recommendation and fertilizer recommendation algorithms.

4.2.2: Integration of GPS for Location

- GPS Sensor Setup: Integrate a GPS module (like NeoGPS) with the microcontroller.
- Location Retrieval: Retrieve real-time latitude longitude coordinates and elevation data from the GPS module.
- Data Processing: Process GPS data to obtain accurate location information.
- Integration: Combine location data with other input variables (NPK, rainfall, humidity) for precise recommendations.

4.2.3: WiFi Module and Arduino for Data Transmission

- Hardware Setup: Connect a Wifi module (esp8266) and an Arduino Uno.
- Data Preparation: Format the collected data (NPK, rainfall, humidity, location) into a structured message.
- Message Sending: Use commands to send the formatted data to a recipient.
- Data Reception: Set up a receiving system on the recipient end to receive and process the transmitted data.
- Integration: Integrate the Wifi module and Arduino for real-time data transmission.

By following these methodologies, you can effectively gather real-time data from NPK sensors, weather APIs, GPS modules, Wifi modules, and Arduino , enhancing the accuracy and relevancy of your crop recommendation and fertilizer recommendation system.

4.3: Algorithms Used

4.3.1: Random Forest Classifier

The Random Forest algorithm is a popular and powerful machine learning technique used for both classification and regression tasks. It belongs to the family of ensemble methods, which combine multiple individual models to make more accurate predictions. Random Forest builds an ensemble of decision trees, where each tree is constructed using a random subset of the training data and a random subset of the input features.The algorithm works by creating a large number of decision trees, typically referred to as the forest. Each decision tree is trained on a different subset of the training data, chosen through a process called bootstrap aggregating or bagging. This means that each tree is exposed to a slightly different set of examples, promoting diversity among the individual models.

During the construction of each decision tree, the algorithm selects a subset of features at random, which helps introduce further variation among the trees. This random feature selection ensures that each tree focuses on different aspects of the data and reduces the risk of overfitting.

To make predictions using the Random Forest, each tree in the ensemble independently generates a prediction, and the final prediction is determined by a majority vote (for classification) or an average (for regression) of the predictions from all the trees. This ensemble approach leads to robust and accurate predictions, as it combines the strengths of multiple models and mitigates the weaknesses of individual trees.

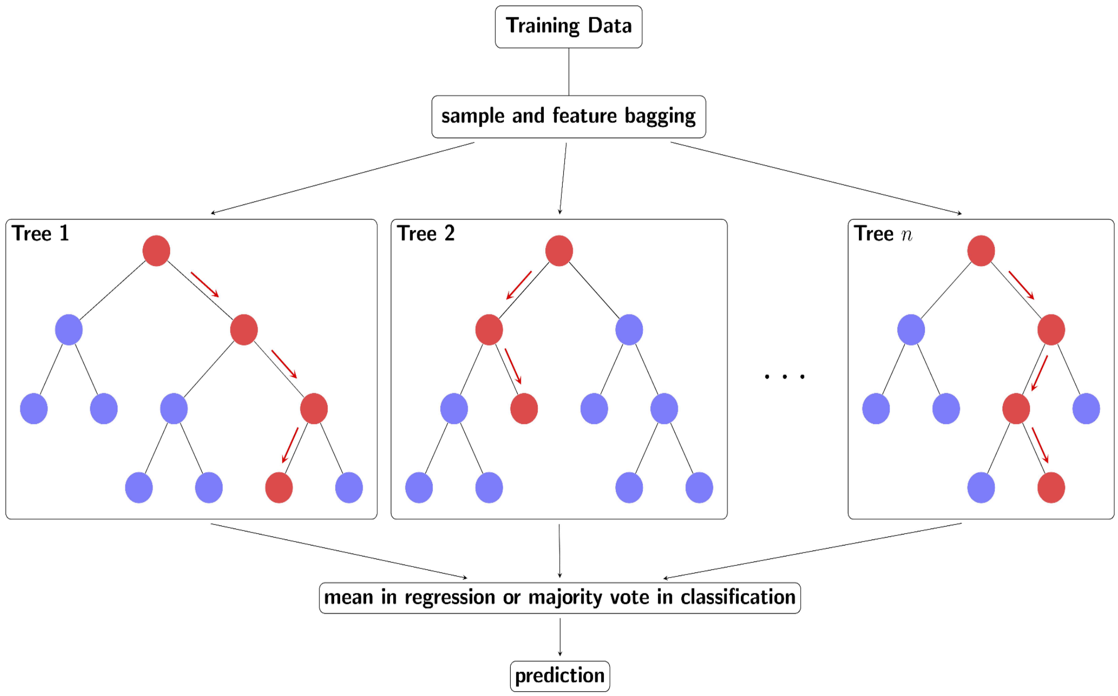

Figure 4.3.1: Random Forest Classifier

[Source:https://tex.stackexchange.com/questions/503883/illustrating-the-random-forest-algorithm-in-tikzAccessed June-8,2023]

4.3.2: Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful and widely used machine learning algorithm that is primarily employed for classification and regression tasks. Its key objective is to create a hyperplane in a high-dimensional feature space, which best separates different classes of data points. The distinctive characteristic of SVM lies in its ability to maximize the margin or the distance between the decision boundary and the nearest data points from each class. By maximizing this margin, SVM aims to achieve better generalization and robustness to new, unseen data. SVM utilizes a subset of training examples called support vectors, which are the data points closest to the decision boundary. These support vectors play a crucial role in defining the decision boundary and determining the classification of new instances. SVM can handle both linearly separable and non-linearly separable datasets by applying the kernel trick, which transforms the input space into a higher-dimensional feature space, where the data becomes separable.

This flexibility enables SVM to capture complex relationships between features and achieve accurate predictions. SVM has been widely adopted in various domains, including image recognition, text categorization, and bioinformatics, due to its effectiveness and versatility in solving a wide range of machine learning problems.

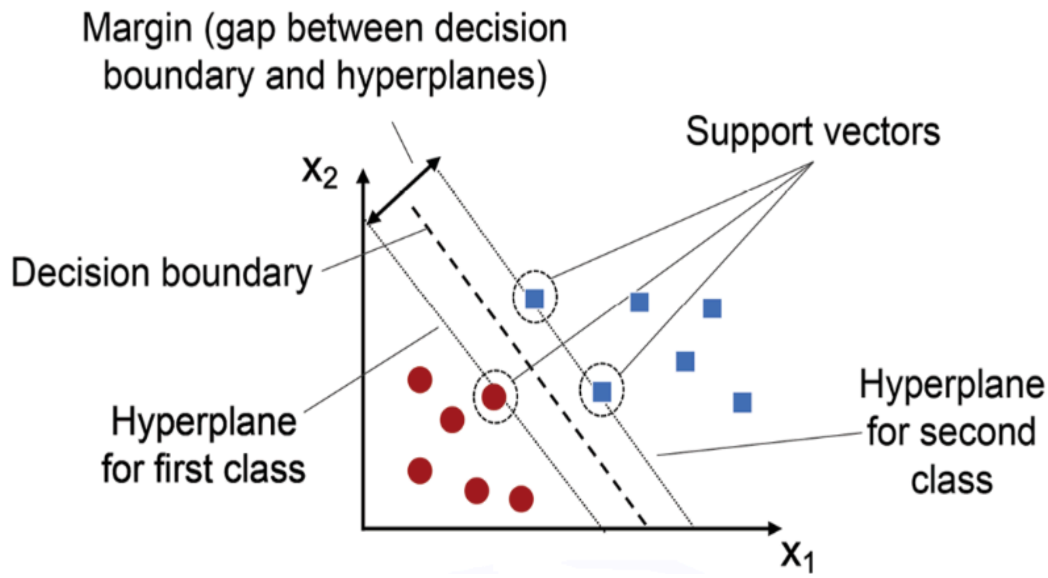

Figure 4.3.2 Support Vector Machine

*[Source:*https://vitalflux.com/wp-content/uploads/2022/08/support-vector-machine-1.png Accessed June-8,2023]*

4.3.3: Decision Tree

A decision tree is a predictive modeling technique that uses a tree-like structure to make decisions or predictions based on a series of hierarchical rules. It is a supervised learning algorithm that can be used for both classification and regression tasks.

In a decision tree, 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 predicted value. The tree structure is constructed by recursively partitioning the data based on the values of the input features.

The process of building a decision tree involves selecting the most informative attribute at each node as the splitting criterion. Various measures, such as information gain, Gini index, or entropy, can be used to determine the best attribute that maximizes the separation of the classes or reduces the impurity in the nodes.

The splitting continues until a stopping criterion is met, such as reaching a maximum depth, having a minimum number of samples in a node, or when further splits do not improve the classification accuracy significantly.

Once the decision tree is constructed, it can be used to make predictions for new instances by following the path from the root to a leaf node based on the attribute tests. In the case of classification, the class label associated with the leaf node is assigned to the instance. In regression, the predicted value is often the average or majority value of the training instances in the leaf node.

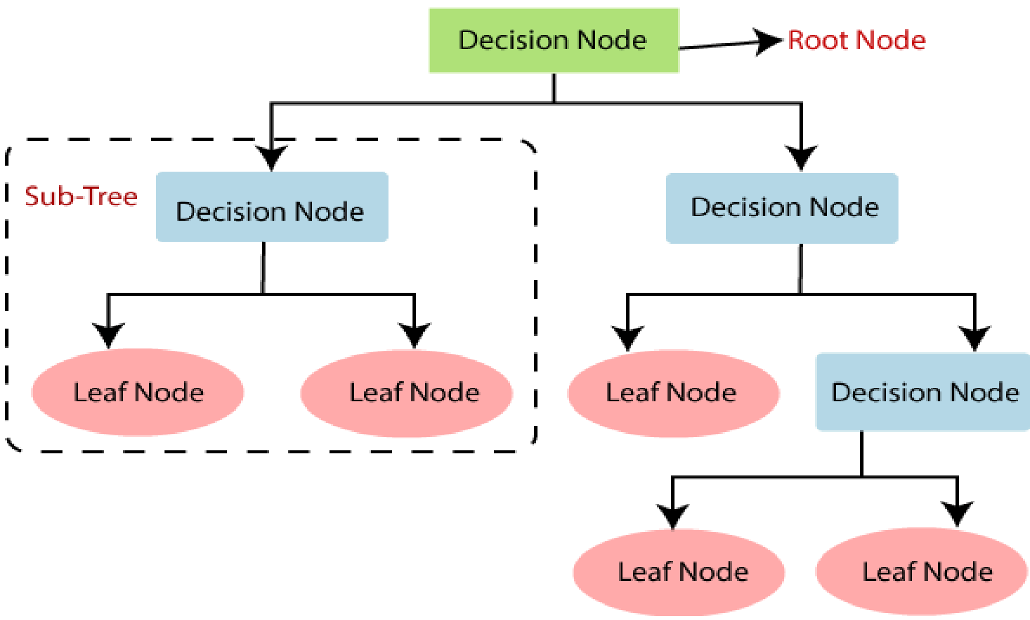

Figure 4.3.3 Decision Tree

[Source:https://www.devops.ae/wp-content/uploads/2021/04/decision-tree-classification-algorithm.png Accessed June-28,2023]

4.3.4: Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a specific type of ANN. It is a feedforward neural network that consists of multiple layers of artificial neurons (also known as perceptrons) that are connected in a directed acyclic graph. Each layer processes the information received from the previous layer and passes it on to the next layer until the output is generated . MLPs are commonly used for supervised learning tasks such as classification and regression.[10]

There are typically three types of layers in an MLP:

- Input Layer: The input layer is the first layer in an MLP and it receives the inputs for the network. The inputs can be any type of data, such as images, text, or numerical values. The input layer does not perform any computations.
- Hidden Layers: Hidden layers are located between the input layer and the output layer and are responsible for processing the inputs and generating intermediate representations of the data. These intermediate representations are then passed on to the next layer for further processing. The number of hidden layers in an MLP can vary depending on the complexity of the problem being solved.
- Output Layer: The output layer is the last layer in an MLP and it produces the final outputs for the network. The outputs can be class labels in a classification problem, continuous values in a regression problem, or any other type of information that the network was designed to produce.

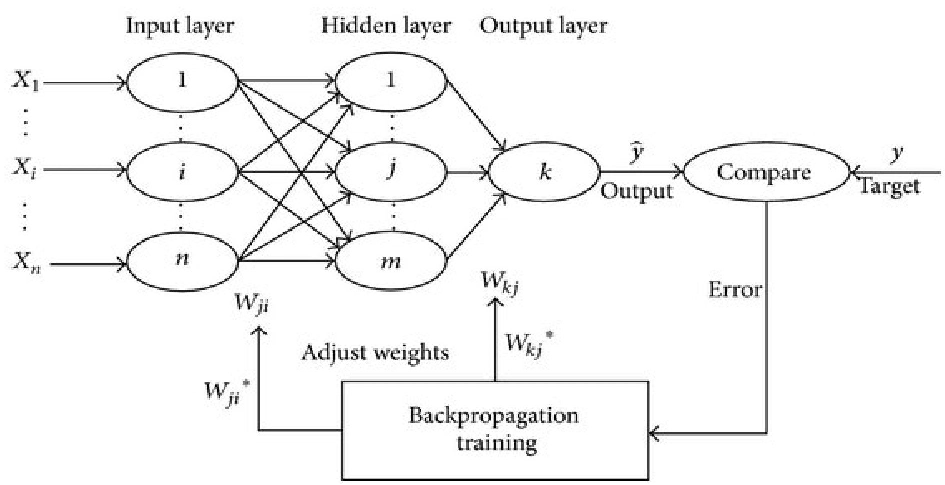

Figure 4.3.4 Multilayer Perceptron (MLP)

[Source:https://eadn-wc03-4064062.nxedge.io/cdn/wp-content/uploads/2020/12/back propagation-training-algorithm.jpg

Accessed June-28,2023]

CHAPTER 5: SYSTEM DESIGN AND ARCHITECTURE

5.1: Block Diagram

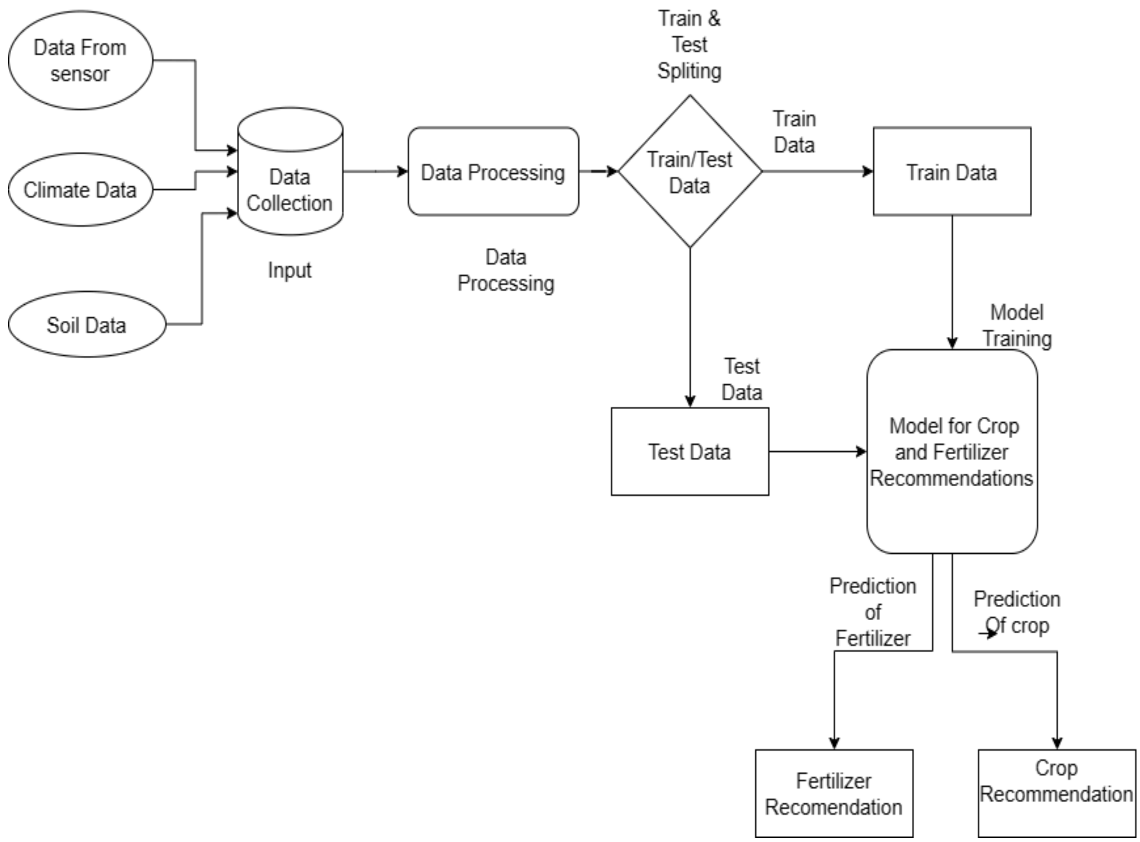

Figure 5.1: Block diagram of crop and fertilizer recommendation

5.2: DFD Diagram

5.2.1: DFD Level 0

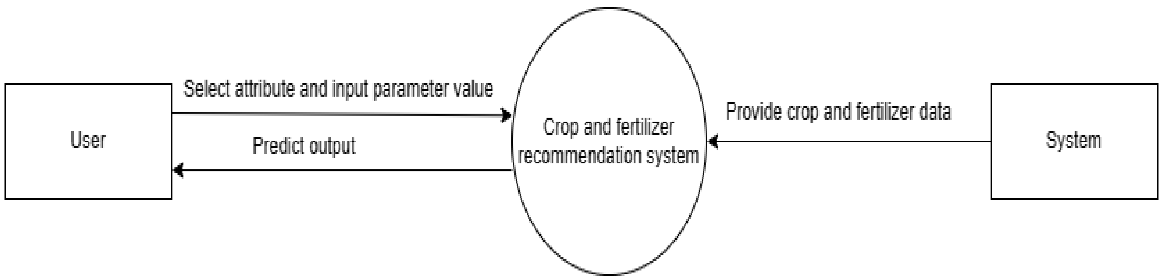

Figure 5.2.1: DFD level 0 diagram

5.2.2: DFD Level 1

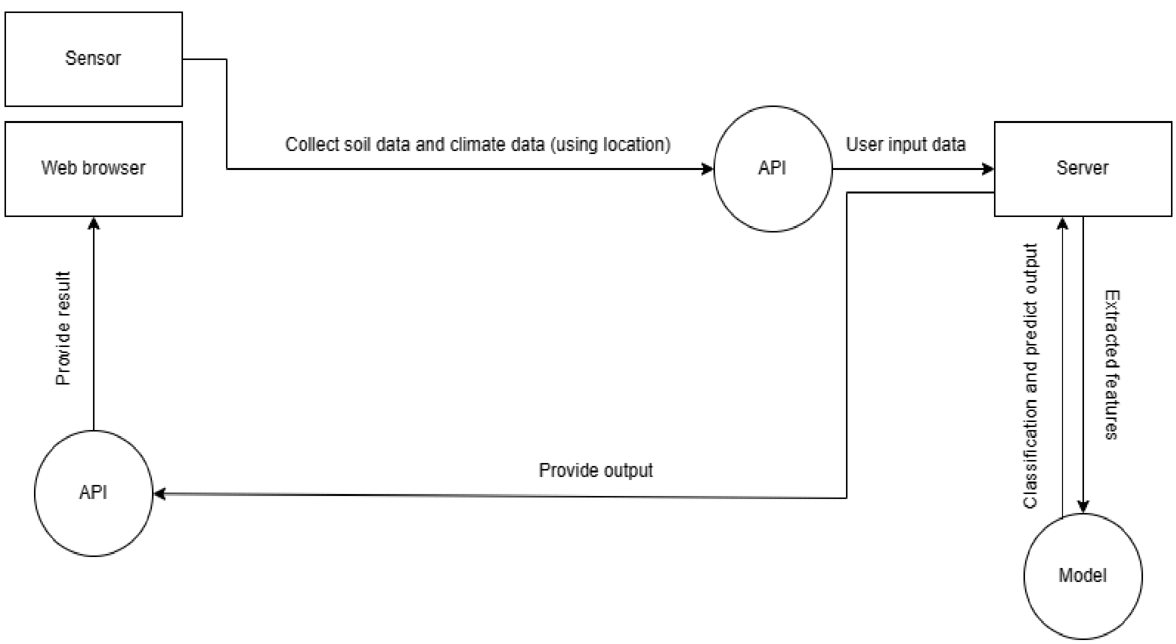

Figure 5.2.2: DFD level 1 diagram

5.3: Use Case Diagram

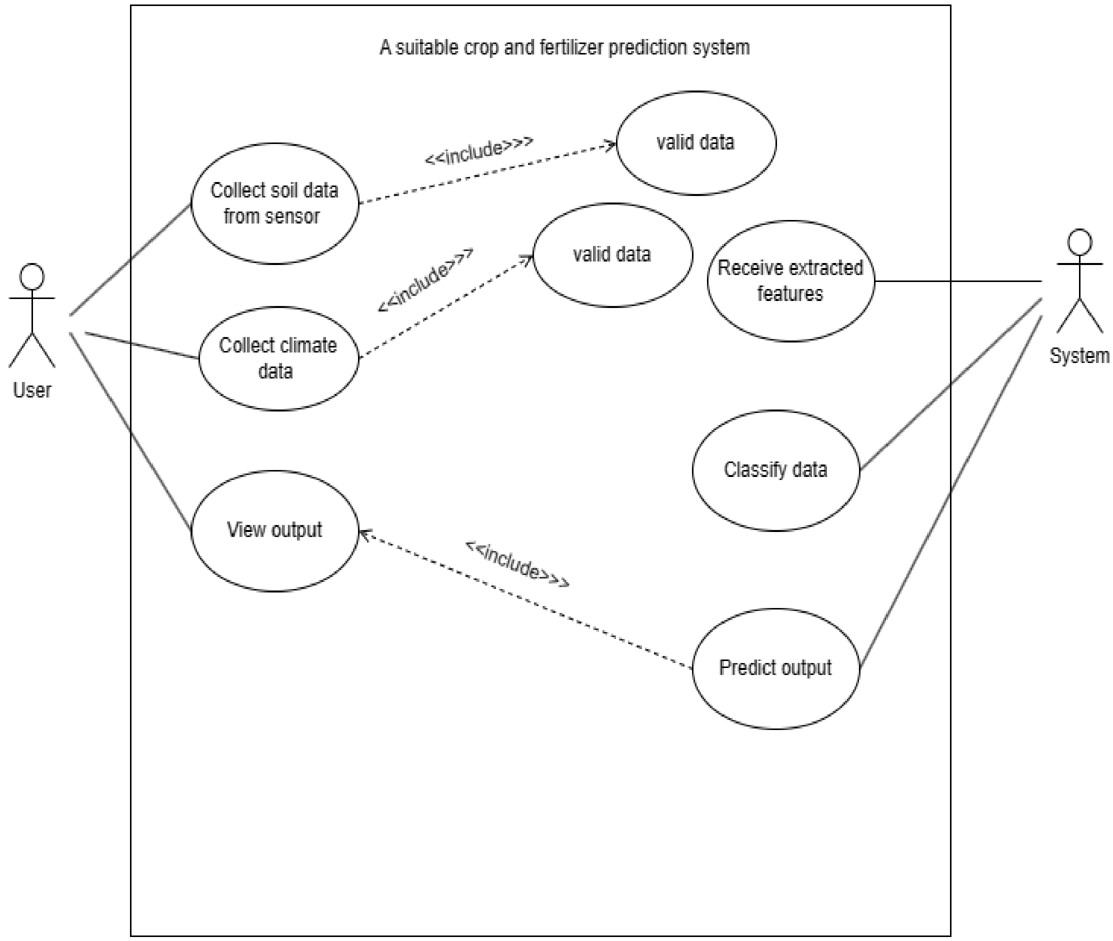

Figure 5.3: Use case diagram

CHAPTER 6: RESULTS AND ANALYSIS

6.1: Results

The model is trained and tested on a diverse dataset and has demonstrated a decent accuracy across various machine learning algorithms. By leveraging comprehensive soil and weather data, our machine learning models enhance the precision of crop recommendations, offering farmers tailored guidance for optimal crop selection. The seamless integration of real-time data from soil and weather sensors ensures that recommendations are based on the latest information, thereby elevating the accuracy and relevance of the guidance provided.

The development of a user-friendly web interface adds to the project's strengths. The interface's simplicity in design and navigation contributes to a positive user experience, making the system accessible and appealing to users with varying levels of technological literacy.

Here are the screenshots of the user interface of this model where the user can see different values sent through the sensor and the recommended crop and fertilizer at the end.

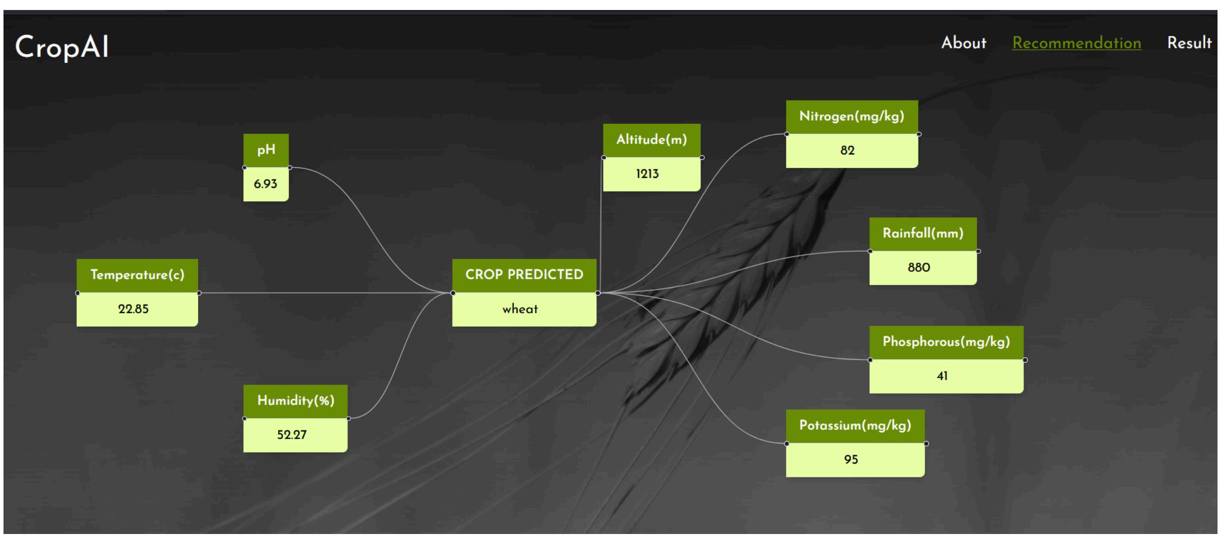

Figure 6.1.1: Web interface showing various sensor inputs

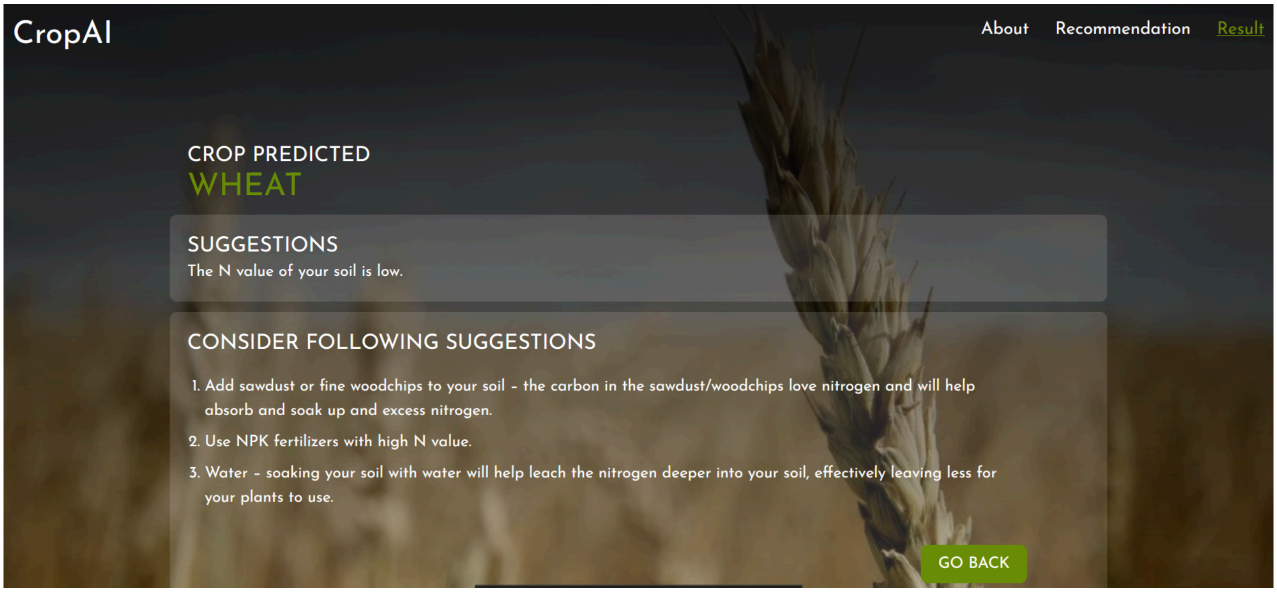

Figure 6.1.2: Web interface showing recommended crop and fertilizer

6.2: Analysis

The models have achieved an overall accuracy of approximately 50%, signifying a respectable predictive performance. As the system undergoes refinement, we anticipate an increase in accuracy, reflecting our commitment to continuous improvement.

While the recommendations are valuable, certain areas need work, including data quality, feature selection, and the system's adaptability to new scenarios. Our ongoing efforts focus on refining the system by adjusting features, fine-tuning settings, and gaining a deeper understanding of decision-making processes. We learned that factors like nitrogen levels and temperature are crucial for accurate predictions.

Initial user feedback suggests that automatic data collection through sensors makes things easier, streamlining the user experience and positioning the system as a practical tool for farmers.

This report provides a snapshot of the project's current status. Regular updates and iterations will contribute to the continuous improvement of the crop recommendation system.

CHAPTER 7: CONCLUSION, LIMITATIONS AND FUTURE ENHANCEMENT

7.1: Conclusion

In conclusion, our crop and fertilizer recommendation system represent a significant step towards applying technology for precision agriculture. The project has demonstrated promising results with commendable accuracy in machine learning algorithms trained on extensive soil and weather datasets. The user-friendly web interface enhances accessibility for farmers, contributing to a positive and practical experience.

The integration of real-time data from soil and weather sensors ensures that our recommendations are founded on the most current information, reflecting our commitment to providing precise guidance for optimal crop selection. While the system has achieved a baseline accuracy of approximately 50%, ongoing refinement efforts are underway to enhance this performance.

7.2: Limitations

Despite the project's successes, certain limitations and challenges have been identified. Areas such as data quality, feature selection, similarity in the range of attributes across different crops and the system's adaptability to diverse scenarios require attention. Addressing these limitations is crucial for optimizing the system's predictive capabilities and ensuring robust performance across varying agricultural conditions.

Additionally, the reliance on historical datasets may introduce biases or fail to capture evolving trends, highlighting the need for continuous updates and improvements to maintain relevance.

7.3: Future Enhancement

This project is just the beginning. We're constantly working to make our crop recommendation system even better and do more. We're building on this project to unlock even more features and accuracy for our crop recommendation system. Key areas for improvement include:

- Advanced Machine Learning Techniques: Explore advanced machine learning techniques, including deep learning approaches, to capture more complex patterns in soil and weather data.
- Dynamic Updating: Implement mechanisms for dynamic updating of the model with real-time data to ensure the system remains adaptive to evolving agricultural conditions.
- User Feedback Integration: Establish a feedback loop with users to gather insights into the practical application of recommendations, enabling continuous refinement based on real-world experiences.
- IoT Expansion: Consider expanding the range of IoT devices and sensors to gather a more comprehensive set of agricultural data, including pest infestations, disease detection, and crop health monitoring.
- Collaboration with Agronomists: Foster collaboration with agronomists and domain experts to enhance the system's understanding of complex agricultural ecosystems and further improve the accuracy of recommendations.

REFERENCES

[1] P.Priya, U.Muthaiah, M.Balamurugan, “Predicting yield of the crop using machine learning Algorithm”, IJESRT et al., 7(4): April-2018, pp 2277-2284. [Accessed June-28 2023]

[2] Georg Ruß, Rudolf Kruse, Martin Schneider, and Peter Wagner. “Estimation of neural network parameters for wheat yield prediction” In Max Bramer, editor, Artificial Intelligence in Theory and Practice II, volume 276 of IFIP International Federation for Information Processing, pages 109–118. Springer, July 2008. [Accessed June-28 2023]

[3] S. Veenadhari, B. Misra and C. Singh, “Machine learning approach for forecasting crop yield based on climatic parameters” 2014 International Conference on Computer Communication and Informatics, Coimbatore, 2014, pp. 1-5.. [Accessed June-28 2023]

[4] Jun Wu, Anastasiya Olesnikova, Chi- Hwa Song, Won Don Lee (2009), “The Development and Application of Decision Tree for Agriculture Data” IITSI, pp 16-20. [Accessed June-28 2023]

[5] Crop Recommendation on Analyzing Soil Using Machine Learning – Anguraj. K, Thiyaneswaran. B, Megashree. G, Preetha Shri. J. G, Navya. S, Jayanthi. J – 2021.

[6] Crop Monitoring and Recommendation System using Machine Learning and IOT – R. Pallavi Reddy, B. Vinitha, K. Rishitha, K. Pranavi – 2020.

[7] Crop Recommendation and Yield Prediction for Agriculture Using Data Mining Techniques – Aakunuri Manjula, Dr. G. Narsimha – 2019.

[8] Crop Recommendation System and Plant Disease Classification using Machine Learning for Precision Agriculture – Mahendra Choudhary, Rohit Sartandel, Anish Arun, Leena Ladge – 2018.

[9] L. C. Gavade, "Detection of N, P, K using Fiber Optic Sensor and PIC Controller," International Journal of Engineering Science,vol. 13787, 2017. [Accessed June-28 2023]

BIBLIOGRAPHY

[a] *NARC*. https://soil.narc.gov.np/. Accessed 9 June 2023.

[b] *Kaggle: Your Machine Learning and Data Science Community*. https://www.kaggle.com/. Accessed 9 Jun. 2023.

APPENDIX A: DATASET

1	Crop	pH	Temperature (°C)	Humidity (%)	Altitude (m)	Rainfall (mm)	Nitrogen (mg/kg)	Phosphorus (mg/kg)	Potassium (mg/kg)
2	Hazelnut	7.45	23.21	69.59	603	717.54	81	40	82
3	Hazelnut	5.61	24.34	78.54	1035	1038.3	113	27	95
4	Hazelnut	5.59	19.24	78.95	1172	903.85	105	27	118
5	Hazelnut	6.86	15.71	68.84	294	754.4	68	37	82
6	Hazelnut	5.93	13.88	69.83	1342	859.22	59	28	113
7	Hazelnut	5.64	10.67	51.73	1341	644.9	69	34	115
8	Hazelnut	6.93	20.24	60.86	1024	721.67	48	37	120
9	Hazelnut	6.34	14.74	72.61	927	928.46	119	33	86
10	Hazelnut	6.73	22.43	54.62	1434	858.34	114	23	101
11	Hazelnut	5.98	13.68	72.73	1326	636.54	47	36	87
12	Hazelnut	7.41	18.97	58.45	1023	646.32	92	26	87
13	Hazelnut	6.02	12.92	63.37	194	624.74	83	36	112
14	Hazelnut	5.95	18.54	75.15	594	827.86	70	33	117
15	Hazelnut	5.7	10.67	63.69	1357	1091.59	115	28	92

Fig 7.1: A snippet of dataset for Hazelnut fruit

The dataset used for training and testing the crop and fertilizer recommendation system consists of a comprehensive collection of eight key attributes for a specific crop. The attributes include:

- pH: The acidity or alkalinity of the soil, influencing nutrient availability.
- Temperature: The ambient temperature of the agricultural environment, affecting crop growth.
- Humidity: The moisture content in the air, influencing plant transpiration and evaporation.
- Altitude: The elevation above sea level, which can impact atmospheric pressure and temperature.
- Rainfall: The amount of precipitation, a critical factor for water availability to crops.
- Nitrogen Content: The concentration of nitrogen in the soil, a vital nutrient for plant growth.
- Phosphorus Content: The concentration of phosphorus in the soil, essential for root development and energy transfer.
- Potassium Content: The concentration of potassium in the soil, crucial for overall plant health and stress resistance.

The dataset is structured to encompass 200 combinations of these eight attributes for each of the 100 crops considered in the recommendation system. This diversity allows the machine learning models to learn the complex relationships between soil, weather conditions, and crop suitability.

The records in the dataset serve as input for training the Random Forest, SVM, MLP, and Decision Tree algorithms. The goal is to enable the models to generalize patterns and make accurate predictions for crop recommendations based on new input data gathered from the sensors in the deployed system.

APPENDIX B: SOURCE CODE

```
void loop() {
  // Read DHT Sensor data
  float humi = dht.readHumidity();
  float tempC = dht.readTemperature();
  if (!isnan(humi) && !isnan(tempC)) {
    StaticJsonDocument<200> jsonDocument;
    jsonDocument["humi"] = humi;
    jsonDocument["tempC"] = tempC;

    String jsonString;
    serializeJson(jsonDocument, jsonString);
    Serial.println(jsonString);
  } else {
    Serial.println("Failed to read from DHT sensor!");
  }
  // Read GPS data
  while (ss.available() > 0) {
    gps.encode(ss.read());
    if (gps.location.isUpdated()) {
      Serial.print("Latitude= ");
      Serial.print(gps.location.lat(), 6);
      Serial.print(" Longitude= ");
      Serial.println(gps.location.lng(), 6);
      Serial.print(" Altitude= ");
      Serial.println(gps.altitude.meters(), 6);
    }
  }
  // Read pH Sensor data
  int measurings = 0;
  for (int i = 0; i < pH_Samples; i++) {
    measurings += analogRead(pHSense);
    delay(10);
  }
  float voltage = 5 / adc_resolution * measurings / pH_Samples;
  Serial.print("pH= ");
  Serial.println(ph(voltage));
```

Fig 7.2: A snapshot of arduino code to read sensors' data

The above Arduino code reads data from various sensors, including a DHT22 temperature and humidity sensor, a GPS module, and a pH sensor. The data is then sent to the local server using ESP8266 Wifi module. Finally, this input is used to recommend the most suitable crop to the user.

```
# Define the accuracies and model names
model_accuracies = [mlp_accuracy, rf_accuracy, dt_accuracy, svm_accuracy]
model_names = ["MLP", "Random Forest", "Decision Tree", "SVM"]

# Print accuracies
for model_name, accuracy in zip(model_names, model_accuracies):
    print(f"{model_name} Accuracy: {accuracy}")

# Plotting
plt.figure(figsize=(10, 6))
plt.bar(model_names, model_accuracies, color=['blue', 'green', 'red', 'purple'])
plt.title('Model Accuracies')
plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.ylim(0, 1)  # Assuming accuracy ranges from 0 to 1
plt.show()
```

```
MLP Accuracy: 0.5961538461538461
Random Forest Accuracy: 0.5677655677655677
Decision Tree Accuracy: 0.47234432234432233
SVM Accuracy: 0.49340659340659343
```

Fig 7.3: A snapshot of code printing accuracies of different models

Our model uses four machine and deep learning algorithms and recommends the crop to the user using the model which has the highest accuracy. The above code prints the accuracies of MLP, Random Forest, Decision Tree and SVM algorithm. So this system compares the accuracies and recommends a crop to the user using the MLP model as it has the highest accuracy among all at this instant.