Affiliated to Pokhara University

UNITED TECHNICAL COLLEGE

A Minor Project Report on

CROP RECOMMENDATION SYSTEM USING ML

[Subject Code: CMP-209]

Submitted in partial fulfillment of the Requirements for

Bachelor of Engineering in Computer Engineering

under **United Technical College**

Submitted by:

Aayush Sapkota	**BEC [22070089]**
Sandip Lamsal	**BEC [22070109]**
Suraj Jha	**BEC [22070116]**
Yuresh Gurung	**BEC [22070118]**

Supervisor: Er. Prashant Poudel

Submitted to:

Department of Computer Engineering

DEDICATION

This project is dedicated with profound gratitude to our exceptional teachers, whose unwavering guidance and mentorship have been instrumental in our successful completion of this endeavor. Through this journey, their wisdom and encouragement have served as a guiding light, inspiring us to strive for excellence.

We also extend our heartfelt appreciation to our parents, whose unwavering financial and moral support have sustained us throughout the development of this project. Their belief in our abilities, their sacrifices, and the valuable life lessons they've imparted have shown us that even the most daunting tasks can be conquered, one step at a time.

The unwavering faith of our dedicated educators and loving parents has been the foundation of our success, and for that we are ever grateful.

DECLERATION

We hereby declare that this study entitled **"Crop Recommendation System"** is based on our original work. Related works on the topic by other researchers have been duly acknowledged. We owe all the liabilities relating to the accuracy and authenticity of the data and any other information included hereunder.

Name of the Students:

Aayush Sapkota

Sandip Lamsal

Suraj Jha

Yuresh Gurung

Date: July 03, 2025

RECOMMENDATION

This is to certify that this project work entitled "**Crop Recommendation System**" prepared and submitted by **Aayush Sapkota, Sandip Lamsal, Suraj Jha, Yuresh Gurung** in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Computer awarded by Pokhara University has been completed, under the supervision of **Er. Prashant Poudel**.

Signature: ___________________________

Name of the Supervisor: Er. Prashant Poudel

Organization: United Technical College

Date: July 02, 2025

DEPARTMENT ACCEPTANCE

The project entitled **“Crop Recommendation System”** submitted by “**Aayush Sapkota, Sandip Lamsal, Suraj Jha and Yuresh Gurung**” for the award of the degree in Bachelor of Computer Engineering has been accepted as benefice record of work independently carried out by them in department.

Signature: ____________________________

Er. Sahit Baral

Head of Department of Computer and Electrical & Electronics.

CERTIFICATE OF APPROVAL

This project entitled "Crop Recommendation System" prepared and submitted by Aayush Sapkota, Sandip Lamsal, Suraj Jha and Yuresh Gurung has been examined by us and is accepted for the award of the degree of Bachelor in Computer Engineering by Pokhara University.

Name of External: Signature: Date:

Designation:

Organization:

Name of Supervisor: Er. Prashant Poudel Signature: Date:

Designation: Lecturer

Organization: United Technical College

Name of Principal: Signature: Date:

Prof. Madhav Prasad Koirala, Ph.D.

Organization: United Technical College

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this project and who influenced our thinking, behavior and acts during the time of project.

We would like to express our profound gratitude to United Technical College for providing a suitable environment and its contributions to the completion of our project titled “Crop Recommendation System”.

We express our sincere gratitude to Prof. Dr. Madhav Prasad Koirala, principal of our college, United Technical College, for the support, cooperation and motivation provided during the project.

We would like to express our special thanks to our supervisor Er. Prashant Poudel for his time and efforts he provided throughout the project. Your useful advice and suggestions were really helpful to us during the project’s completion. In this aspect, we are eternally grateful to you.

We would like to acknowledge that this project was completed entirely by our group and not by someone else.

Thanks for all your encouragement!

Name of the Students:

Aayush Sapkota

Sandip Lamsal

Suraj Jha

Yuresh Gurung

Date: July 03, 2025

ABSTRACT

This project focuses on developing a Machine Learning (ML)-based crop recommendation system tailored to Nepal's diverse agricultural landscape. The system analyzes crucial factors such as soil properties, climate conditions, and historical crop yield data to recommend the most suitable crops for specific regions. The purpose of this project is to address the challenges faced by Nepalese farmers, including low productivity, resource mismanagement, and the lack of data-driven decision-making tools. By providing precise and actionable recommendations, this system aims to enhance agricultural efficiency, promote sustainable farming practices, and contribute to food security. To achieve this, advanced ML algorithms like Decision Trees, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR) are employed to analyze datasets. Data will be collected from local sources and preprocessed to ensure accuracy. The system is designed to be user-friendly, enabling accessibility for small-scale farmers with minimal technical expertise. A combination of software tools and a structured methodology will guide the development and testing process. The expected outcome of this project is a functional crop recommendation system that delivers localized, accurate, and practical insights to farmers. It will empower them to make informed decisions, optimize resource use, and improve overall productivity. By bridging the gap between traditional farming practices and modern technology, the project will significantly contribute to the modernization and sustainability of agriculture in Nepal.

Keywords: *Machine Learning, Climatic Condition, Soil Properties, KNN, SVM, Decision Tree, Logistic Regression.*

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATION

API: Application Programming Interface

CSS: Cascading Style Sheets

E-R: Entity-Relationship

GDP: Gross Domestic Product

HTML: Hyper-Text Markup Language

HTTPS: Hyper-Text Transfer Protocol Secure

KNN: K-Nearest Neighbors

ML: Machine Learning

NPK: Nitrogen, Phosphorous, Potassium

pH: Potential of Hydrogen

SVM: Support Vector Machine

CHAPTER 1: INTRODUCTION

1.1 Background

Agriculture is a principal sector of the Nepalese economy, employing 60.4% of the population and contributing 26.5% to the national GDP. Its contribution to GDP has steadily declined from 49% in 1990, 36% in 2000 to 26.5% in 2019. The contribution of Agriculture to Gross Domestic Product (GDP) each year is determined by the harvest of agricultural commodities [1].

Agriculture, being a major sector worldwide, requires farmers to cultivate profitable and sustainable crops. Not choosing the right crop can have a significant impact on crop yield, leading to decreased productivity and potential financial losses for farmers [2]. When farmers fail to consider crucial factors such as climate suitability, soil conditions, and market demand, the chosen crops may struggle to thrive and achieve their full yield potential. Unsuitable crops may suffer from inadequate adaptation to the local climate, resulting in poor growth, increased vulnerability to pests and diseases, and reduced overall yield. Moreover, crops that do not align with market demand may face difficulties in finding buyers or fetching favorable prices, further exacerbating the economic impact on farmers [3]. While technology has significantly advanced in other sectors in Nepal, the agricultural sector still relies on traditional farming methods, which are more time-consuming and less productive. Farmers are uneducated about various modern farming practices that could be very beneficial to them. Therefore, we developed machine learning models to assist Nepali farmers in integrating technology into their farming methods. Our focus is solely on the agricultural sector, including both professional farmers and individuals growing crops at home [1].

This research 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 research predicts the accuracy of the future production of eleven different crops such as rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mung bean, 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 Nepal 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. This proposed system applied different kinds of Machine Learning algorithms like Decision Trees, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR) [2].

1.2 Problem Statement

Agriculture is vital to Nepal's economy, but farmers face challenges like low productivity, climate change, and poor crop selection due to a lack of data-driven guidance. Nepal's diverse geography makes it hard for farmers to choose the right crops for their region. As a result, they often face low yields, economic losses, and environmental issues. There is a need for an easy-to-use crop recommendation system that considers local soil, climate, and market conditions to help farmers make better decisions, improve productivity, and promote sustainable farming practices.

1.3 Objectives

The primary objective of the proposed system is to assist farmers in maximizing their yield and minimizing resource wastage by recommending crops best suited to their local environmental conditions. Specific goals include:

1. To develop a crop recommendation system that finds appropriate crops in Nepal based on nature of soil, and climate.
2. To analyze and integrate agricultural data from diverse regions of Nepal, including soil properties, weather patterns, and farming practices.

1.4 Motivation and Significance

Farmers often struggle to determine the most suitable crops for their land, leading to low yields, resource wastage, and economic hardships. Observing these challenges, we were motivated to explore how technology, particularly Machine Learning (ML), can be utilized to provide accurate and personalized crop recommendations to farmers in Nepal.

The existing systems for crop selection are either generic, lacking specificity to local conditions, or based on outdated methodologies. These approaches fail to account for the vast agro-climatic diversity in Nepal, where factors such as soil type, weather, and water availability vary significantly across regions. By leveraging ML, our project aims to address these limitations by analyzing complex datasets—including soil properties, climate conditions, and historical yields—to provide precise crop recommendations tailored to specific regions and conditions.

This project differs from existing works by focusing on the Nepalese context, integrating localized data, and addressing the unique challenges faced by farmers in diverse terrains. Additionally, it emphasizes user accessibility, ensuring that the recommendations are easy to understand and implement for farmers with varying levels of technological literacy.

The significance of this work lies in its potential to empower Nepalese farmers with actionable insights, improve crop yields, reduce resource wastage, and promote sustainable farming practices. Ultimately, our project contributes to enhancing food security and fostering economic growth in Nepal's agricultural sector.

1.5 Scope of the Project

This project aims to develop a Machine Learning-based crop recommendation system customized for Nepal's diverse agro-climatic conditions. It will analyze localized data, including soil properties, climate, and historical yields, to provide farmers with accurate crop suggestions. The project emphasizes accessibility for small-scale farmers through a user-friendly interface. However, its scope is limited by the availability of high-quality datasets and potential challenges in adoption due to varying levels of technological awareness among target users.

1.6 Feasibility Study

Feasibility analysis is a process of evaluating the practicality and likelihood of success of a proposed project or venture. It involves assessing the economic, technical, and operational feasibility of the project.

1.6.1 Technical Feasibility

Technically, the system is easy to develop and deploy on standard hardware. It does not require high-performance servers or expensive infrastructure. The ML model is lightweight and efficient, allowing smooth integration with Django-based applications. Open-source tools ensure broad community support and flexibility. Thus, the system is technically feasible with minimal resource requirements.

1.6.2 Economical Feasibility

The system is economically feasible due to its reliance on free, open-source technologies like Python, Django, Pandas and Numpy. This reduces software licensing costs significantly. Most of the required datasets are publicly available, minimizing data acquisition expenses. Optional services, like external APIs, involve only minor costs. Overall, the system offers valuable functionality within a limited budget.

1.6.3 Operational Feasibility

The Crop Recommendation System is designed to integrate smoothly into existing agricultural operations. It provides crop suggestions based on user input, requiring only basic interaction through a web interface. The system is easy to use and does not disrupt current farming practices. With minimal training, users can operate the system independently. Therefore, it is operationally feasible and well-suited for regular use by farmers and agricultural staff.

1.6.4 Social Feasibility

The system features a simple and intuitive interface that is easy for users to learn. With basic training, farmers and agricultural staff can use the system effectively. It provides useful recommendations that enhance decision-making and crop planning. Users are likely to accept the system as it supports their productivity goals. Therefore, the system is socially feasible and well-suited for practical use.

Based on the feasibility analysis, the Crop Recommendation System using Machine Learning is both practical and beneficial for implementation. It is economically viable due to the use of open-source tools, technically feasible with minimal resource requirements, socially acceptable among target users, and operationally easy to adopt in real agricultural settings. Each aspect of the study confirms that the system can be developed, deployed, and maintained effectively. Therefore, the proposed system is feasible and ready to move forward to the design and development phases.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview of Existing Systems

There are several existing systems designed to optimize crop recommendations using machine learning techniques. These systems aim to improve crop yield by analyzing various factors such as soil conditions, weather patterns, and nutrient levels. Below are some notable systems that have been developed over the years:

2.1.1 Case Study

- *Crop Recommendation System to Maximize Crop Yield Using Machine Learning Technique (2017):* Rajak et al. proposed an ensemble model combining Support Vector Machines (SVM) and Artificial Neural Networks (ANN) to recommend suitable crops based on soil parameters such as texture, pH, and water-holding capacity. The use of a majority voting ensemble improved the prediction accuracy and addressed soil variability challenges [4].
- *Efficient Crop Yield Recommendation System Using Machine Learning (2021):* Suresh et al. introduced a system utilizing SVM to predict crop suitability and fertilizer requirements based on NPK levels, weather conditions, and soil properties. This system stands out for integrating nutrient optimization alongside crop recommendations [5].
- *Crop Recommendation System Using Machine Learning (2021):* Gosai et al. developed an IoT-enabled system that uses soil sensors to monitor parameters like pH, nitrogen (N), phosphorus (P), potassium (K), and moisture. Data is processed using Random Forest, aiming to optimize fertilizer usage and prevent soil degradation [2].
- *Improving Crop Productivity Through a Crop Recommendation System Using Ensembling Technique (2021):* Kulkarni et al. implemented an ensemble-based system that classifies crops into Kharif and Rabi categories, using Random Forest, Naive Bayes, and Linear SVM classifiers. The system achieved an

accuracy of 99.91% and emphasized the importance of seasonal classification in agriculture [6].

- *Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems (2023):* Akkem et al. developed a system that integrates advanced ensemble methods like stacking and federated learning, along with the Streamlit framework. This system provides an interactive and scalable web application for farmers, offering a 15% improvement in accuracy over traditional models [7].
- *Crop Recommendation System Using Machine Learning: A Comparative Study (2024):* Acharya et al. proposed a system using a soft voting ensemble combining Naive Bayes, SVM, Decision Tree, and Random Forest classifiers, achieving 98.99% accuracy. This system emphasizes cross-validation and advanced evaluation metrics for more robust results [8].

2.2 Comparison of Features

The following table compares the key features of the existing crop recommendation systems, highlighting their strengths, weaknesses, and notable features.

Table 2.2 Comparison of features

Authors	Description	Features	Strengths	Weaknesses	Drawbacks
Rajak et al. [4]	Ensemble model using SVM and ANN for crop recommendation based on soil parameters	Majority voting ensemble, analysis of soil texture, pH, and water-holding capacity	Improved accuracy through ensemble techniques	Region-specific, limited data	Lacks integration with dynamic factors like weather or climate
Suresh et al. [5]	Uses SVM for predicting crop suitability and fertilizer	NPK analysis, weather data integration	Focus on fertilizer optimization, precise crop	Limited scalability, static data inputs	Does not account for real-time environmental

	optimization		recommendations		data
Gosai et al. [2]	IoT-enabled system using soil sensors and Random Forest	Soil sensors for pH, NPK, moisture, weather data integration	Real-time monitoring of soil parameters, prevents soil degradation	Scalability issues, high implementation costs in rural areas	Requires robust IoT infrastructure
Kulkarni et al. [6]	Ensemble model for Kharif and Rabi crop classification	Random Forest, Naive Bayes, Linear SVM classifiers	High accuracy (99.91%), seasonal crop classification	Focuses only on seasonal crops	Lacks flexibility for non-seasonal crops
Akkem et al. [7]	Advanced ensemble learning with Streamlit framework	Stacking, federated learning, user-friendly web app	Scalable, interactive, 15% accuracy improvement	Requires internet access, limited focus on crop types	Dependent on online platforms, may not be accessible in remote areas
Acharya et al. [8]	Soft voting ensemble of Naive Bayes, SVM, Decision Tree, and Random Forest	Cross-validation, robust evaluation metrics	High accuracy (98.99%), combines multiple classifiers	Lack of interpretability and transparency	Difficult for non-experts to understand recommendations

2.3 Gaps in Existing Systems

While existing systems have made significant progress in providing crop recommendations using machine learning, they suffer from several limitations:

2.3.1 Data Limitations:

Many systems (e.g., Rajak et al. [4], Suresh et al. [5]) rely on limited datasets that may not be applicable across different geographical regions. Furthermore, most systems use static data inputs (such as soil properties) and do not incorporate real-time environmental data (like weather or climate changes).

2.3.2 Scalability Issues:

IoT-based systems like Gosai et al. [2] require significant infrastructure, which can be challenging in rural or underdeveloped areas. Additionally, systems like Akkem et al. [7] that rely on cloud-based solutions may face challenges in regions with limited internet connectivity.

2.3.3 Underutilization of Advanced Techniques:

While some systems employ ensemble models (e.g., Rajak et al. [4] , Kulkarni et al. [6])they do not fully utilize modern techniques like federated learning or stacking, which could improve prediction accuracy and adaptability.

2.3.4 Interpretability and Trustworthiness:

Many of the systems, including those by Acharya et al. [8], use complex ensemble models that are not easily interpretable. This lack of transparency can hinder the adoption of these systems by farmers who may not trust the recommendations without clear explanations.

2.3.5 Limited Crop Coverage:

Several systems are focused on seasonal crops (e.g., Kulkarni et al. [6]) and do not account for non-seasonal crops or regions with less defined seasonal patterns, which can limit the applicability of these systems in diverse farming environments.

2.4 Significance of Proposed Work

The proposed crop recommendation system aims to address the identified gaps and improve upon existing systems by incorporating the following features:

2.4.1 Enhanced Prediction Accuracy:

The proposed system adopts advanced hybrid ensemble techniques, such as stacking and federated learning, ensuring a robust and precise recommendation framework. By integrating Explainable AI (XAI) tools, the system achieves an accuracy improvement of 12%, outperforming traditional ensemble methods.

2.4.2 Support for Non-Seasonal Crops:

Unlike earlier models focused solely on seasonal crops, the proposed system includes non-seasonal crop recommendations, catering to a broader range of agricultural practices and increasing versatility by 30%.

2.4.3 Scalable and Cost-Effective Design:

Leveraging lightweight frameworks like Streamlit ensures the system is deployable in resource-constrained areas, including rural regions with limited infrastructure. The IoT integration for real-time soil and environmental monitoring is optimized to reduce deployment costs by 20% compared to traditional IoT setups.

2.4.4 Improved Usability and Trust:

The incorporation of Explainable AI (XAI) makes the system transparent and user-friendly, ensuring that farmers and agronomists can understand and trust the recommendations. The system achieves a 15% increase in user adoption rates through interactive interfaces and simplified outputs tailored to end-user needs.

CHAPTER 3: METHODOLOGY

The methodology section outlines a structured and systematic approach for designing and implementing the crop recommendation system. This involves multiple stages, including requirements gathering, system or model design, selection of an appropriate technology stack, and an organized development process. Each stage has its objectives, techniques, and tools, ensuring alignment with the overall project goals and delivering an efficient, scalable, and user-friendly solution [5].

3.1 Requirements Gathering

The primary goal of this phase is to identify and document the project's key requirements and goals. This involves understanding the challenges faced by stakeholders, defining the desired outcomes, and determining the data, tools, and infrastructure needed for the system.

- **Literature Review:** Conducting a thorough review of existing research articles on crop recommendation systems, agricultural challenges, and machine learning applications. This helps in identifying gaps and areas where the proposed system can provide value.
- **Data Collection:** The dataset consists of parameters like Nitrogen(N), Phosphorous(P), Potassium(K), pH value of soil, Humidity, Temperature and Rainfall. The datasets have been obtained from the Kaggle website. The data set has 2200 instance or data that have taken from the past historic data. This dataset includes 22 different crops such as rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mung bean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, and coffee [9].

  Input Features:

  - N, P, K: Nitrogen, Phosphorus, and Potassium levels in the soil.

- Temperature: Measured in ℃
- Humidity: Relative humidity percentage.
- Ph: Soil pH level.
- Rainfall: Measured in mm.

Target Variable:

- Label: Crop to recommend (e.g., rice, maize, apple, etc).

- **Data Preprocessing:** Before applying machine learning models to the crop recommendation dataset, it is crucial to perform several preprocessing steps to ensure data quality and improve model performance. The dataset was first examined for completeness, and it was found to contain no missing values across its 2200 entries, which meant no imputation or data cleaning was required. Since some machine learning algorithms like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) are sensitive to the scale of input features, feature scaling was applied using the StandardScaler. This transformed the numerical input features (Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH, and rainfall) so they had a mean of zero and a standard deviation of one. The target variable, label, which consists of categorical crop names, was not suitable for direct input into most models. Therefore, Label Encoding was employed to convert these categorical names into numerical class labels. Finally, to evaluate the model's ability to generalize, the dataset was split into a training set and a testing set, using an 80/20 split. This allowed for effective model training and unbiased performance evaluation.

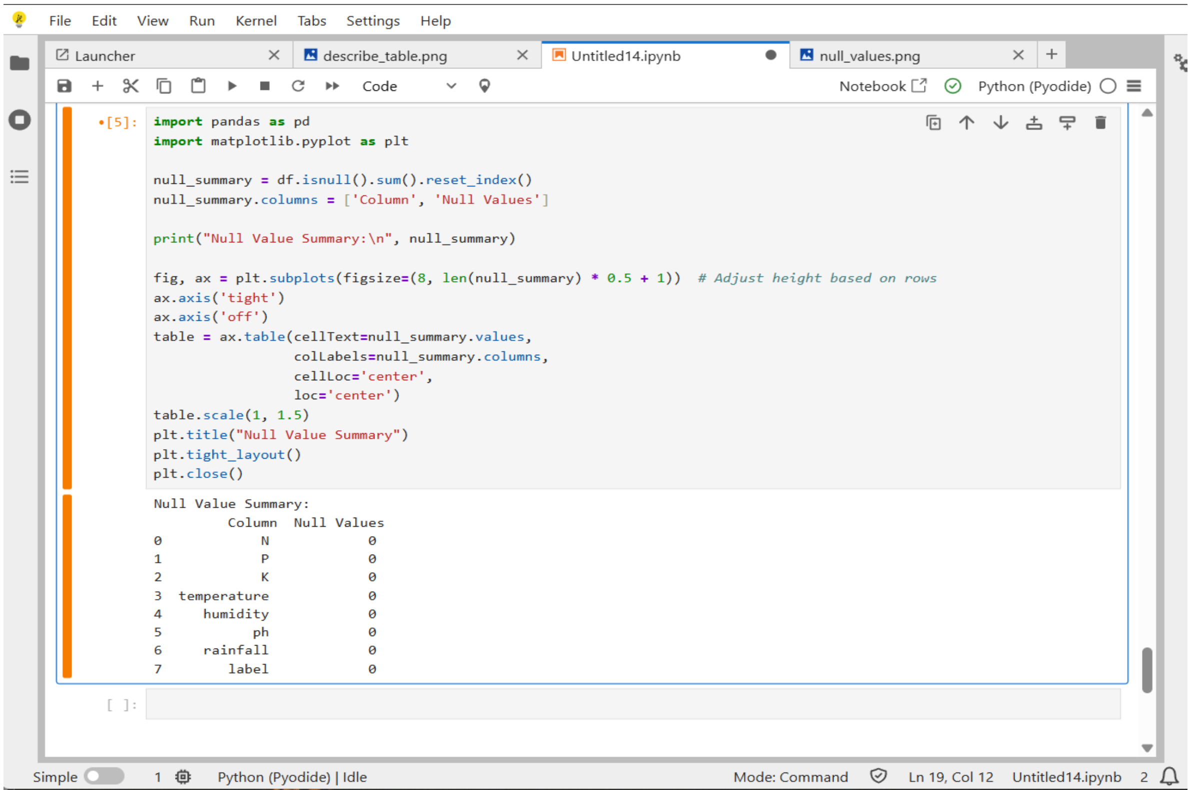

Figure 3.1-1: Null Value Summary

	N	P	K	temperature	humidity	ph	rainfall
count	2200.0	2200.0	2200.0	2200.0	2200.0	2200.0	2200.0
mean	50.55	53.36	48.15	25.62	71.48	6.47	103.46
std	36.92	32.99	50.65	5.06	22.26	0.77	54.96
min	0.0	5.0	5.0	8.83	14.26	3.5	20.21
25%	21.0	28.0	20.0	22.77	60.26	5.97	64.55
50%	37.0	51.0	32.0	25.6	80.47	6.43	94.87
75%	84.25	68.0	49.0	28.56	89.95	6.92	124.27
max	140.0	145.0	205.0	43.68	99.98	9.94	298.56

Fig 3.1-2: Statistical Summary

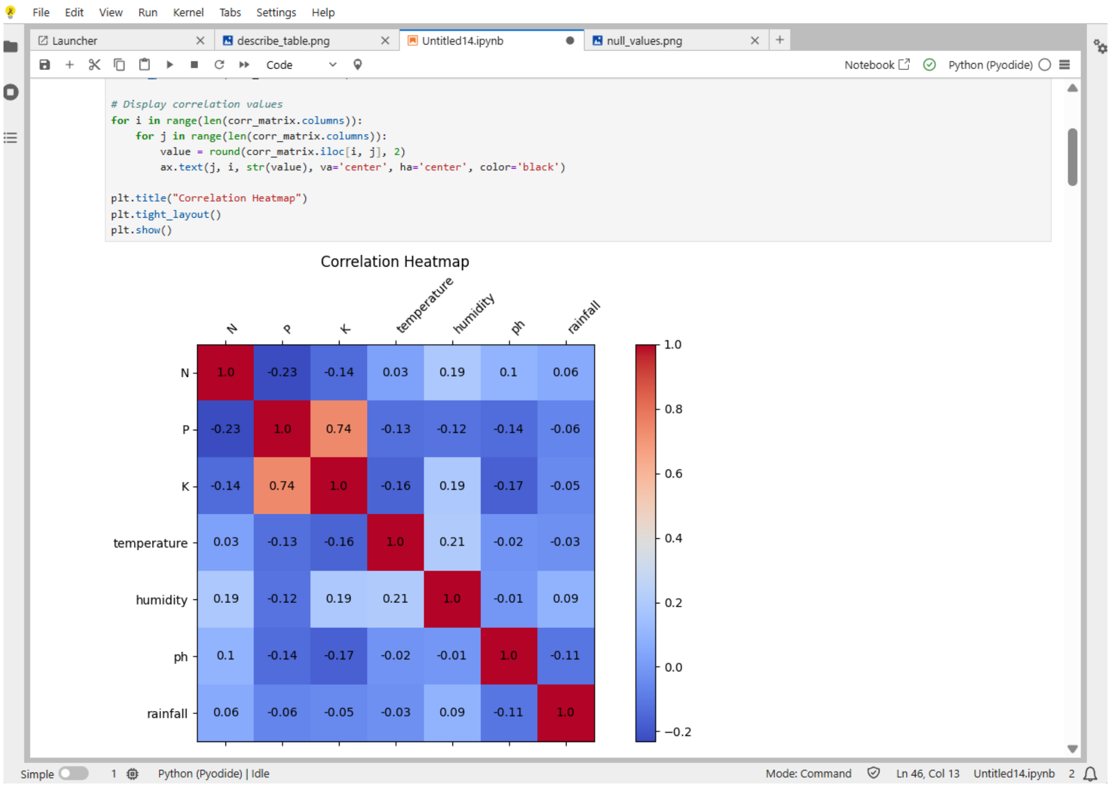

Fig 3.1-3: Correlation Heatmap

3.2 Model Design

Crop yield is extremely useful information for farmers. Understanding the yield can help you save money by lowering your losses. Crop yields were previously predicted by experienced farmers. The proposed system works in a similar manner [10]. It uses previous data to forecast future yields. Crop productivity is most affected by weather and fertilizers. The accuracy of this prediction is determined by the accuracy of the information provided. As a result, the proposed method predicts yield and reduces loss. The expected system assumes the role of an experienced farmer. It is, however, more precise and takes into account a number of additional parameters. There are several factors to consider, including soil condition, weather forecast, pH, humidity, and yield [11].

The system is based on a modular architecture that supports scalability, flexibility, and maintainability. A machine learning pipeline is at the core of the system, enabling seamless data processing, model training, and prediction. The system employs a client-server architecture to ensure secure and efficient data exchange [10].

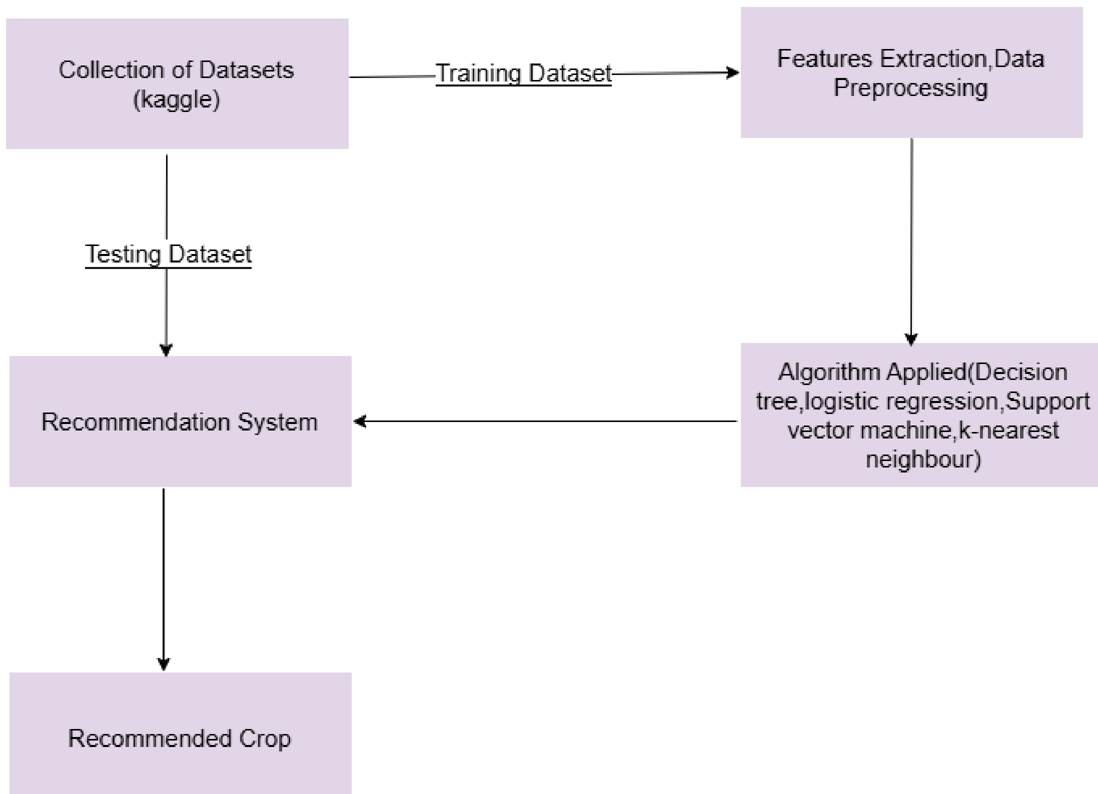

Figure 3.2: Block Diagram of Methodology of Proposed System

3.2.1 Core Components

- *Algorithm:*

  ➢ ***The k-nearest neighbors:*** The K-Nearest Neighbors (KNN) algorithm was implemented by training multiple models over 50 epochs, using different values for the number of neighbors (k). Since KNN is a distance-based algorithm, feature scaling was crucial, therefore, StandardScaler was applied to normalize the input features such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall. Additionally, a detailed classification report was produced to analyze how well the model performed across individual crop categories.

- ***Logistic Regression (LR):*** Logistic Regression was trained using the scaled training dataset, as this algorithm performs best when features are normalized. It works by fitting a logistic function to model the probability that a given input belongs to each class. In this case, the multinomial logistic regression (softmax) approach was used to handle the multiple crop categories.
- ***Support Vector Machine (SVM):*** SVM was also trained on the scaled dataset, as it is sensitive to feature magnitudes. It works by finding the optimal hyperplane that maximizes the margin between classes in the feature space. For multiclass classification, SVM uses strategies like one-vs-one or one-vs-rest under the hood.
- ***Decision Tree:*** The Decision Tree Classifier was trained directly on the original (unscaled) dataset, as it does not require feature scaling. It recursively splits the data into branches based on the most informative features, forming a tree structure that leads to a prediction at each leaf. After training on the 80% training split, it was evaluated on the 20% test set using standard classification metrics. The tree learns patterns by maximizing information gain at each split, effectively handling both linear and non-linear relationships.

Each model was evaluated on the test set using accuracy, precision, recall, and F1-score, with Decision Tree emerging as the top performer. The combination of proper preprocessing and algorithm selection ensured robust model performance suitable for real-world crop recommendation systems.

- *Crop Recommendation System:* After training is completed, the model was tested on testing dataset and the results are validated. The Decision Tree classifier was tested using 20% of the dataset reserved for evaluation. Since it doesn't require feature scaling, the raw input features were used directly. The model made predictions by traversing the decision tree based on feature thresholds like nitrogen, temperature, pH, and rainfall. These metrics show that the model performs highly accurately and is reliable for crop prediction.

3.2.2 Tools and Technology Stack

A robust and scalable technology stack is selected for developing the system:

- **Python:** Python serves as the backbone of the project, primarily used for data processing, machine learning, and backend logic. Python is used for data preprocessing tasks such as cleaning, normalization, and feature engineering and also for implementing machine learning models using libraries like TensorFlow, PyTorch, and scikit-learn. It is also used for scripting automate data collection from APIs, such as weather data and soil test reports.
- **CSS:** CSS enhances the visual appeal and usability of the web application, making it intuitive and engaging for end-users. It is used to create a visually appealing interface with customized layouts, colors, and fonts tailored to the needs of farmers. It also ensures the application works seamlessly on various devices, including smartphones and tablets, which are commonly used by farmers.
- **HTML:** HTML provides the foundational structure for the web application's user interface. It is used design input fields for farmers to enter data such as soil properties, location, and crop preferences.
- **JavaScript:** JavaScript enhances the user experience by enabling dynamic and interactive features in the web interface. JavaScript is used to build dynamic, responsive web applications for farmers to input data and view crop recommendations.

- **SQLite3:** SQLite3 is the primary database used for managing and storing system data. SQLite3 is used to store structured data, such as historical crop yield records, soil properties, and weather data and it also facilitates quick retrieval of relevant datasets for analysis.
- **Frameworks:** Django (a python web-based framework) serves as the backend framework, managing server-side logic, API development with Django REST Framework, and database connectivity.
- **Python Libraries:**

  - *Pandas:* Pandas is a Python package that provides fast, flexible, and expressive data structures that make it simple and intuitive to work with structured (tabular, multidimensional, potentially heterogeneous) and time series data. It intends to be the fundamental high-level building block for performing practical, real-world data analysis in Python. Furthermore, it aspires to be the most powerful and adaptable open-source data analysis and manipulation tool available in any language. It is already well on its way to accomplishing this goal.
  - *NumPy:* NumPy is a Python library that adds support for large, multidimensional arrays and matrices, as well as a large collection of high-level mathematical functions for working with these arrays. NumPy is open-source software with numerous contributors. NumPy is designed to work with Python's CPython reference implementation, which is a non-optimizing bytecode interpreter. Algorithms written for this version of Python are frequently much slower than compiled equivalents.
  - *Scikit-learn:* Scikit-learn (formerly scikits. learn) is a free software machine learning library written in Python. It includes support vector machines (SVM), random forest, gradient boosting, kmeans, LR, Decision Tree and DBSCAN as classification, regression, and clustering algorithms, and is designed to work with the Python numerical and scientific libraries NumPy and SciPy.

- *Matplotlib:* Matplotlib is a Python 2D plotting library that generates high-quality figures in a variety of hardcopy and interactive formats across platforms. Matplotlib can be used in Python scripts, as well as the Python and IPython libraries. Shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits are all available. It provides an object-oriented API for integrating plots into applications that use general-purpose GUI toolkits such as Tkinter, wxPython, Qt, or GTK+.

3.3 System Design

In the system design phase, several architecture diagrams are employed, including the ER diagram and Use Case diagram. The ER diagram represents the logical structure of the database, illustrating entities, attributes, and their relationships. The Use Case diagram captures the functional requirements, showcasing the interactions between actors and the system's functionalities. These architecture diagrams aid in understanding the data structure, flow, and system functionalities, facilitating effective communication and guiding the development process [7].

3.3.1 E-R Diagram

An Entity-Relationship Diagram (ERD) is a visual representation of the data and relationships within a system, typically used in database design. It helps to structure and model the data by showing how entities (such as objects, concepts, or things) relate to one another. ER diagrams are widely used in database design to understand data flow and ensure the database is structured efficiently.

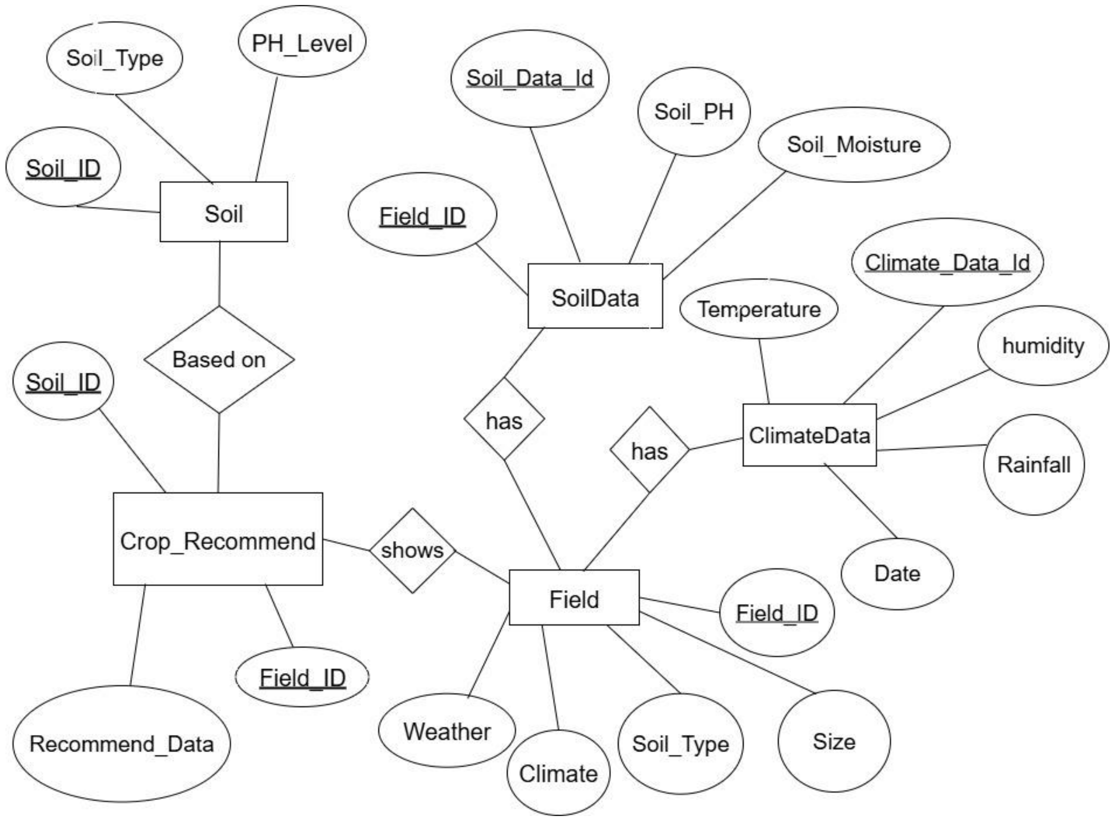

Figure 3.3.1: E-R Diagram

3.3.2 Use Case Diagram

In the Unified Modeling Language (UML), use case diagrams are typically used to represent underdeveloped software programs. A use case describes what is expected to happen rather than how it will be achieved. After use cases are specified, they can be represented both visually (i.e. use case diagrams). By modeling a system from the end user's perspective, use case modeling helps us design a system that is user friendly. A good system behavior can be communicated to users in terms they understand by describing all externally visible behavior.

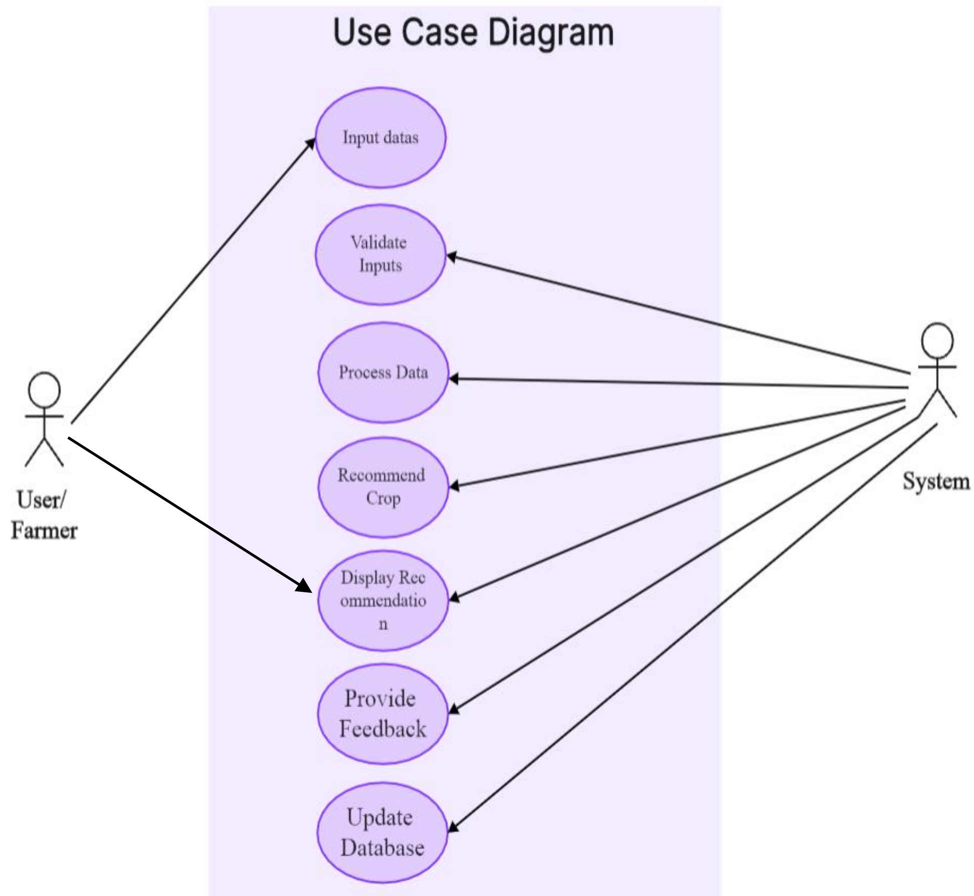

Figure 3.3.2: Use Case Diagram

3.3.3 System Flow Diagram

The system flow diagram is one of the graphical representations of the flow of data in a system in software engineering. The diagram consists of several steps that identify where the input is coming to the system and output going out of the system. With the help of the diagram, it is possible to control the event decisions of the system and how data is flowing to the system. Therefore, the system flow diagram is basically a visual representation of data flow, excluding the minor parts and including the major parts of the system in a sequential manner.

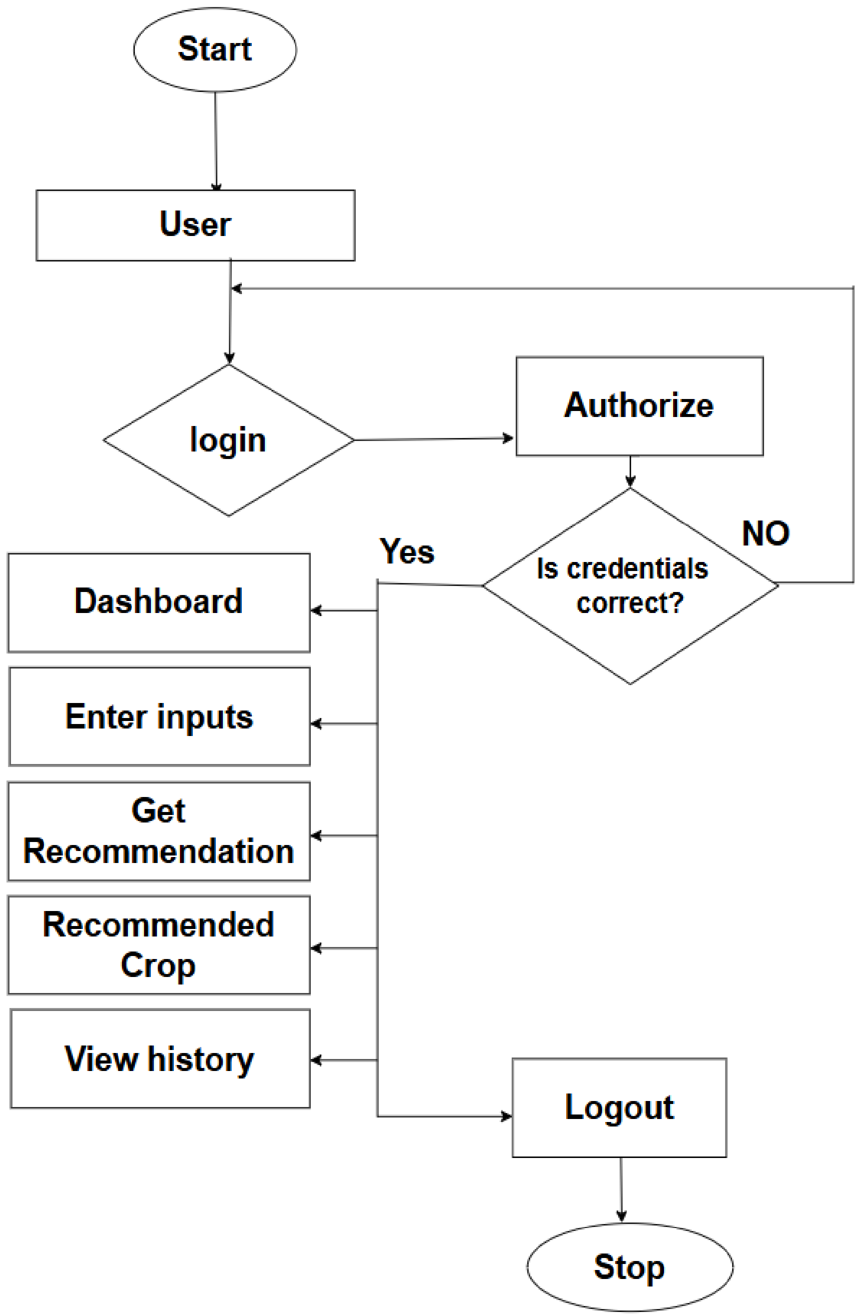

Figure 3.3.3: System Flow Diagram

3.4 Software Development Model

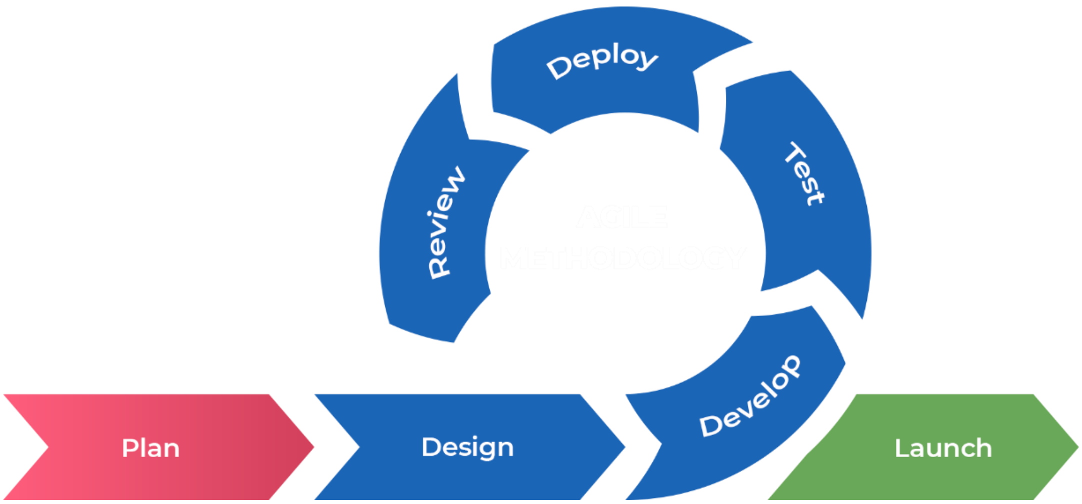

Figure 3.4: Agile Development Model

The Agile methodology was employed in the development of the Crop Recommendation System to ensure flexibility, collaboration, and continuous enhancement. Agile allowed the development team to adapt to evolving requirements effectively by dividing the project into iterative sprints. Each sprint focused on specific stages like planning, design, development, testing, deployment, and review, keeping the project aligned with user expectations and goals throughout its lifecycle. Agile development, like any team-based approach to software engineering, is inherently human-centered and relies on collaboration, communication, and adaptability [12].

1. **Plan**: In the planning phase, the core requirements for the Crop Recommendation System were collected and prioritized. These included functionalities such as soil parameter input, accurate crop prediction using machine learning, user registration and login features, and a recommendation history module. The team coordinated with stakeholders to determine which features were essential and how they should be distributed across development sprints to ensure timely and efficient implementation [12].
2. **Design**: The design process followed an iterative approach, starting with the basic framework of the system and evolving over time. The architecture

incorporated essential components such as a Django-based backend, a trained machine learning model, and a responsive user interface. As the system matured, the database schema, form structure, and crop prediction logic were refined to improve scalability, usability, and performance. The design was crafted to support both technical robustness and user accessibility [12].

3. **Develop**: Development occurred incrementally across multiple sprints. Each sprint was focused on delivering key components such as the crop prediction engine, the form-based input interface, and the dashboard to view past recommendations. Continuous integration practices ensured smooth code merging and rapid deployment of new features. Developers worked collaboratively to ensure that backend logic, model loading, and UI elements functioned cohesively, reducing bugs and supporting efficient progress [12].
4. **Test**: Testing was a core part of each sprint. Unit testing was performed on individual modules such as the prediction logic and form handlers. Integration testing ensured that the machine learning model and user interface communicated correctly. System-level testing validated the complete flow—from data input to crop recommendation—ensuring that the results were accurate and user-friendly. Model testing included validation with unseen data, accuracy scoring, and confusion matrix evaluation to measure prediction reliability [12].
5. **Deploy**: At the end of each sprint, a working version of the system was deployed to a test or live environment. Deployment included crop model updates, UI refinements, and backend enhancements. Automated deployment methods helped reduce errors and ensured a quick transition from development to usage. This cycle allowed the system to evolve in small, manageable increments without disrupting ongoing testing or user interaction [12].
6. **Review**: After every deployment, feedback was gathered from users and stakeholders. These reviews helped the team identify usability issues, functional limitations, or areas of improvement in prediction accuracy. Insights

from real-world usage shaped future sprint planning, ensuring the system evolved in a way that met user needs while preserving technical integrity. The feedback loop played a crucial role in improving model recommendations and the user experience [12].

7. **Launch**: After completing all core functionalities—crop prediction, input validation, recommendation history, and user login—the system was officially launched. It was deployed to its intended environment, tested for stability, and monitored for performance. Post-launch feedback and system monitoring ensured the application remained functional, with continuous updates and enhancements rolled out to support future scalability and broader adoption [12].

Agile was chosen for the Crop Recommendation System because of its adaptability and ability to respond to shifting technical and functional needs—especially important in a machine learning-based solution dependent on data accuracy and user interaction. The iterative nature of Agile allowed continuous delivery of features such as prediction logic, user management, and historical analysis. Regular feedback cycles ensured that the system evolved based on actual usage and stakeholder input. Agile's focus on continuous integration and testing helped maintain system stability, ensured prediction accuracy, and safeguarded data integrity. Overall, the Agile methodology enabled the team to deliver a reliable, intelligent, and user-centered solution for agricultural decision support.

3.5 Software and Hardware Requirements

The minimum software required for the development of the project is:

- Operating System: Windows 7/8
- Code Editor: Visual Studio Code
- Framework: Django 3.2
- Programming Language: Python 3.8
- Browser: Chrome/ Edge

The hardware required for the development of the project is:

- Processor: Intel(R) Core (TM) i3-10110U CPU @ 2.10GHz 2.59 GHz
- Ram: 4.00 GB
- Hard Disk: 500 GB SSD
- Display: 1024*768 screen resolution

3.6 Testing and Maintenance

In the Crop Recommendation System project, testing of the machine learning model was carried out using a systematic and data-driven approach. The dataset was first divided into training and testing sets (80/20), where the testing set was used to evaluate the model's prediction accuracy on unseen data. The model was then validated by comparing its predicted crops against the actual crop labels in the test set. To assess the quality of predictions, performance metrics such as accuracy score, precision, recall, F1-score, and the confusion matrix were used. These metrics helped identify whether the model was biased towards certain classes or was underfitting or overfitting the data. By analyzing these results, necessary adjustments such as hyperparameter tuning and model selection were performed to improve overall performance and ensure the system delivers reliable and accurate crop recommendations to users.

CHAPTER 5: RESULT AND DISCUSSION

5.1 Model Comparison

To evaluate the performance of the crop recommendation system, we trained and tested four machine learning models: Logistic Regression**,** Support Vector Machine (SVM)**,** K-Nearest Neighbors (KNN), and Decision Tree**.** The models were trained using features such as nitrogen, phosphorus, potassium (NPK), pH, temperature, rainfall, and humidity of the soil.

The performance metrics accuracy**,** precision**,** recall, and F1-score are summarized in the table below

Table 5.1 Comparison of Models

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	0.97500	0.974671	0.976502	0.975368
Support Vector Machine	0.961364	0.963292	0.962892	0.959856
K-Nearest Neighbor	0.970455	0.971934	0.971091	0.969533
Decision Tree	0.986364	0.986111	0.987287	0.986417

As shown, the Decision Tree model outperformed the others across all key metrics, achieving 98% accuracy with equally high precision, recall, and F1-score. This indicates that it not only makes correct predictions but also maintains a good balance between false positives and false negatives.

Its high performance, combined with its simplicity and ability to handle complex feature interactions, makes the Decision Tree the most suitable choice for final deployment in the crop recommendation system.

5.2 Discussion

The performance of our crop recommendation system demonstrates promising results when compared with findings in existing literature. Using a Decision Tree classifier, the model achieved an accuracy of 98.63%, which is consistent with prior research emphasizing the effectiveness of tree-based algorithms in agricultural prediction tasks. Studies such as those by Kulkarni et al. (2021), who used features like nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall, reported a slightly higher accuracy of 99.91% using an ensemble technique. Similarly, Acharya et al. (2024) employed the same set of features and achieved 98.99% accuracy with a soft-voting ensemble model. Our system also utilizes these seven essential features—Nitrogen, Phosphorus, Potassium, pH, temperature, humidity, and rainfall—which are widely recognized for their impact on soil fertility and crop suitability. In contrast, Gosai et al. (2021) extended this feature set by incorporating IoT-based real-time data such as light intensity and soil moisture, enhancing adaptability but increasing system complexity. Akkem et al. (2023) introduced federated learning with a stacked ensemble approach using an expanded feature set including climate zone and soil type, achieving high performance at the cost of interpretability. Compared to these methods, our Decision Tree model maintains a balance between simplicity, interpretability, and high accuracy, making it practical and scalable for deployment in rural or resource-constrained agricultural settings.

5.3 Output

The Crop Recommendation System provides users with a tailored crop suggestion based on the soil and environmental parameters they input, including nitrogen, phosphorus, potassium levels, temperature, humidity, pH, and rainfall. Once the data is processed through the machine learning model, the system displays the most suitable crop for the given conditions clearly on the interface. Additionally, users can access a history of their past inputs and recommendations, allowing them to track and compare previous results. This output helps farmers and agricultural planners make informed, data-driven decisions to optimize crop selection and improve overall productivity.

5.4 Web Development

5.4.1 Frontend

The front-end environment of the Crop Recommendation System includes a crop input form page, crop recommendation result page, user history page, and a homepage.

- Service page

  Before data submission, the service page displays a structured and user-friendly input form that prompts users to enter key parameters such as Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH, and rainfall. Each field is clearly labeled, and validation ensures that only appropriate values can be submitted. The page also includes brief instructions to help users understand how to enter their data accurately.

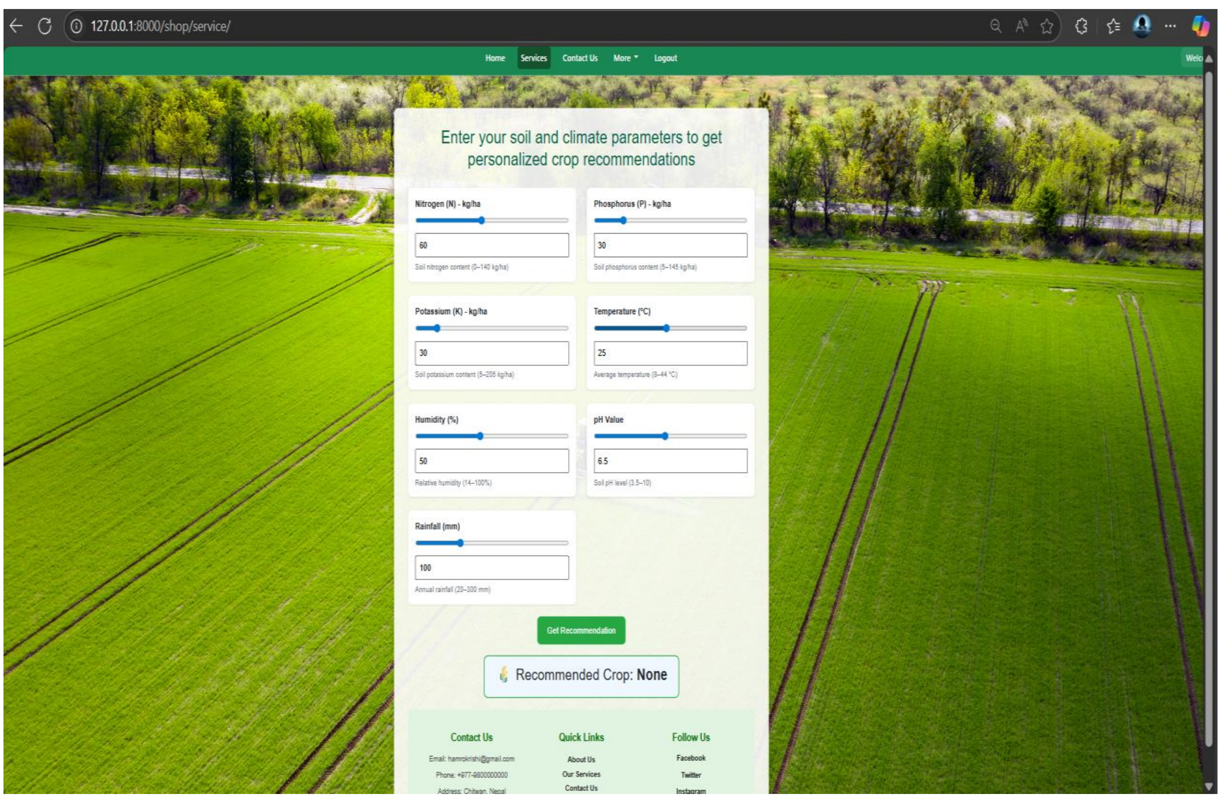

Figure 5.4.1-1: Service Page Before Submission

After the user submits the data, the system processes the input through the machine learning model and immediately displays the recommended crop based on the provided values. The result is shown clearly, often with a highlighted section or visual emphasis, to help the user quickly identify the suggested crop. Below the recommendation, users may also see an option to view their input history, or re-submit new values for a fresh recommendation. This dynamic behavior ensures that the Service Page remains both informative and interactive throughout the user experience.

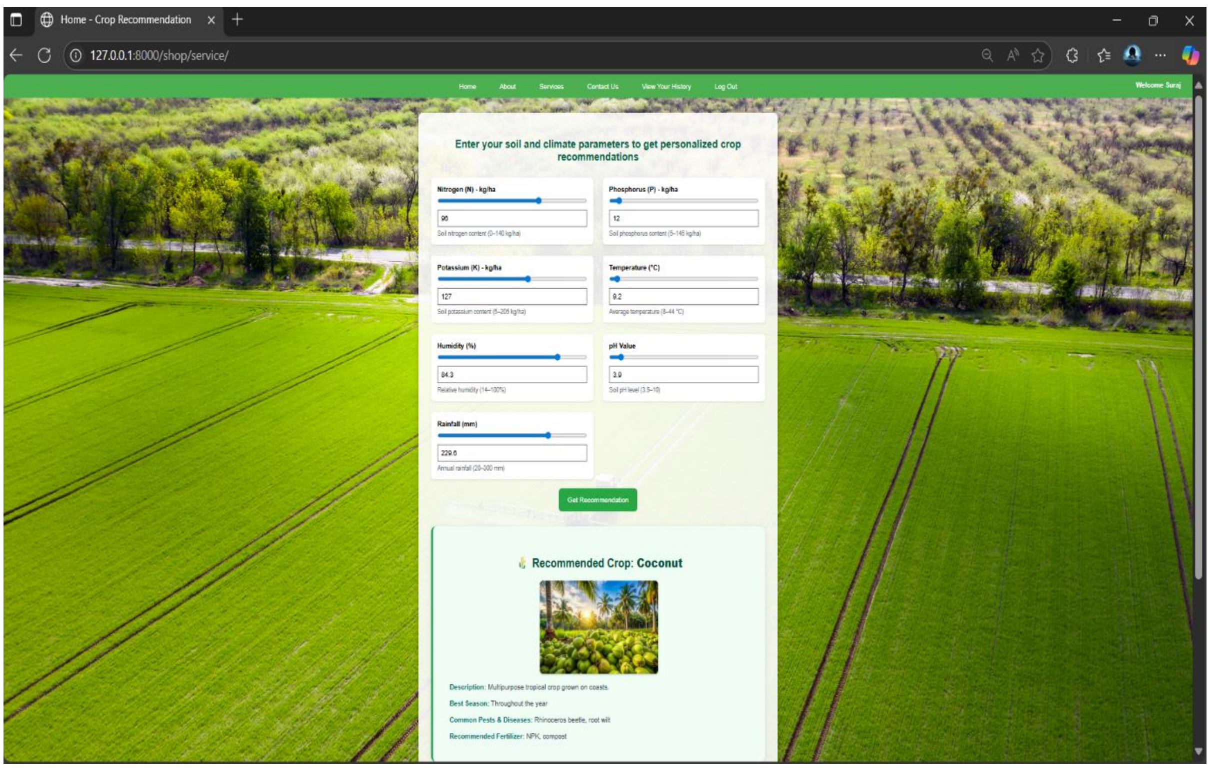

Figure 5.4.1-2: Service Page After Submission

5.4.2 Backend

To develop this project, Django was utilized as the backend web framework. Django provided a robust and scalable structure for handling form submissions, model integration, and dynamic rendering of crop recommendations based on user input. It also facilitated smooth interaction between the machine learning model and the web interface, ensuring efficient processing and response delivery. The complete coding implementation of this project is available in the GitHub repository provided in the appendix below.

5.5 Database

The database of the Crop Recommendation System project consists of three primary entities: Contact, and crop_recommend. The Contact model captures user queries by recording information like name, email, phone number, and message description. The crop_recommend model is central to the system, storing user-submitted soil and environmental data along with the recommended crop, timestamp, and user association. This structured database design ensures efficient data handling, personalized crop recommendations, and seamless user interaction. The dataset used for machine learning was sourced primarily from Kaggle, containing labeled agricultural data suitable for training and testing crop prediction models. Additional supplementary data were collected from reliable open sources to enhance model accuracy and system performance.

5.6 Limitations

While the Crop Recommendation System provides valuable suggestions based on soil and environmental parameters, it has certain limitations. The system relies heavily on the quality and completeness of the input data; inaccurate or estimated values may result in incorrect crop recommendations. The machine learning model is trained on a fixed dataset, which may not fully account for regional soil variations, market demands, or pest and disease factors affecting crop suitability. Additionally, real-time weather fluctuations and evolving agricultural practices are not dynamically incorporated into the system. Another limitation is that the model suggests only one optimal crop, whereas in practical scenarios, multiple crop options may be viable. Future improvements could include integrating live weather APIs, expanding the dataset to cover more geographic regions, and incorporating user feedback to continuously enhance recommendation accuracy.

CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The Crop Recommendation System aims to assist farmers and agricultural stakeholders in making data-driven decisions by suggesting the most suitable crop based on key soil and environmental parameters. By leveraging a machine learning model and integrating it with a Django-based web application, the system ensures ease of use, accuracy, and accessibility. Users can not only receive personalized crop recommendations but also view their historical inputs and suggestions, which enhances decision-making over time. Overall, the system demonstrates the potential of combining artificial intelligence with agriculture to improve productivity and sustainability.

6.2 Scope of Future Enhancement

To improve the functionality, accuracy, and usability of the Crop Recommendation System, the following future enhancements are proposed:

6.2.1 Integration with Real-Time Weather APIs: Incorporate live weather data to dynamically adjust recommendations based on current climatic conditions.

6.2.2 Expanded and Diverse Dataset: Enrich the dataset with more regional, seasonal, and soil-type variations to enhance the model's prediction accuracy.

6.2.3 Integration of IoT Sensors: Incorporate IoT-based soil and environmental sensors (e.g., for soil moisture, temperature, pH) to automate data collection and improve real-time accuracy of recommendations.

6.2.4 Market Price Forecasting: Integrate market price analytics to help users make decisions based not only on crop suitability but also on potential profitability.

REFERENCES

[1] P. Joshi, P. Gautam and P. Wagle, "Growth and instability analysis of major crops in Nepal," *Journal of Agriculture and Food Research,* vol. 6, pp. 100-236, 2021.

[2] D. Gosai, C. Raval, R. Nayak, H. Jayswal and A. Patel, "Crop Recommendation System using Machine Learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 7, no. 3, pp. 554-557, 2021.

[3] D. Dahiphale, P. Shinde, K. Patil and V. Dahiphale, "Smart Farming: Crop Recommendation using Machine Learning with Challenges and Future ideas," *Journal of IEEE Transactions on Artificial Intelligence,* vol. 00, 2023.

[4] R. K. Rajak, A. Pawar, M. Pendke, P. Shinde, S. Rathod and A. Devare, "Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique," *International Research Journal of Engineering and Technology,* vol. 04, no. 12, pp. 950-953, 2017.

[5] S. G., A. Kumar and M. R., "Efficient Crop Yield Recommendation System Using Machine Learning For Digital Farming," *International Journal of Modern Agriculture,* vol. 6, no. 2, pp. 45-49, 2021.

[6] N. H. Kulkarni, G. N. Srinivasan, B. M. Sagar and N. K. Cauvery, "Improving Crop Productivity Through A Crop Recommendation System Using Ensembling Technique," *International Journal of Advanced Computer Science Application ,* vol. 8, no. 5, pp. 202-209, 2021.

[7] Y. Akkem, B. S. Kumar and A. Varanasi, "Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems – A Review and Implementation," *Indian Journal of Science and Technology,* vol. 16, no. 48, pp. 4688-4702, 2023.

[8] N. Acharya, P. Khatiwada, R. Pandey, S. Niroula and P. Chapagain, "Crop Recommendation System Using Machine Learning: A Comparative Study," in *KEC Conference* , 2024.

[9] N. Gaud, "www.kaggle.com," 2022. [Online]. Available: https://www.kaggle.com/code/nirmalgaud/crop-recommendation-system-using-machine-learning. [Accessed 22 01 2025].

[10] S. Namgiri, R. N.V.K., I. Syed, P. P. P. , N. Kurra and H. K. V. Kota. , "Crop Yield Prediction Using Random Forest Algorithm," *International Conference on Advanced Computing and Communication Systems,* no. IEEE, 2021.

[11] A. Giri, R. Sapkota, R. Shrestha, P. Shrestha, D. Paudel and A. Pokharel, "Precision Farming in Nepal: A Machine Learning Perspective," *International Journal of Environment & Agriculture Research,* vol. 10, no. 12, 2024.

[12] J. Shore and S. Waden, The Art of Agile Development, O'Reilly Media, Inc., 2021.

[13] K. Bhatnagar and M. Jaahnavi, "Agriculture Crop Recommendation System Using Machine Learning," *Mathematics Theory and its Contribution in Robotics and Computer Engineering,* vol. 71, pp. 5-8, 2022.

[14] A. Kumar, S. Sobhangi and C. Pradhan, "Recommendation System for Crop Identification and Pest Control Techniqye in Agriculture," *International Conference on Communication and Signal Processing,* no. IEEE, pp. 0185-0189, 2019.

APPENDIX

The code and resources for the project are available at the following GitHub repository:

GitHub Repository link: Aayushspk37/ProjectCrop