A Project On

Fasal Mitra

Smart Farming Using Machine Learning

Completed Successfully

In persuasion for the award of degree Of **BACHELOR OF TECHNOLOGY**

Under the guidance of Asst. Prof. Ranjana Ray and Honourable Dr. Moumita Pal (HOD)

Presented By:

Shivani Kumari	123211002066
Aman Kumar	123211002009
Sayan Sen	123211002064
Shubham Kumar	123211002067

Department of Electronics and Communication

JIS College of Engineering

2025

JIS College of Engineering
Campus : Block 'A', Phase - III, Kalyani,Nadia 741235
Phone : 2582 2137, Telefax : 2582 2865
Email : info@jiscollege.org

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project titled

"Fasal Mitra-SMART FARMING USING MACHINE LEARNING"

Submitted By:

Shivani Kumari 123211002066
Aman Kumar 123211002009
Sayan Sen 123211002064
Shubham Kumar 123211002067

Of BTech 8th Semester, in partial fulfilment of the requirement for the degree of **Bachelor of Technology** in Electronics & Communication Engineering from JISCE (Autonomous Institute), Kalyani, during the academic year 2024-25 is their original Endeavour carried out under my supervision and guidance and has not been presented anywhere else.

…..………………..

HOD of ECE Dept.

JISCE, Kalyani

……………………

Miss. Ranjana Ray

(Project Supervisor)

ACKNOWLEDGMENT

First, we would like to give our sincere gratitude to **Dr. Moumita Pal** and **Miss. Ranjana Ray** for their supervision of our project and for the guidance they provided in shaping our research on **smart agriculture using machine learning**. Their mentorship played a key role in helping us understand the depth and real-world application of AI in farming. Special thanks to **Mr. Anirban Ghosal** for his help and support.

We are also grateful to the entire faculty of the **Electronics and Communication Engineering Department**, including the **Head of the Department**, for their continuous encouragement and mentorship throughout the development of this project.

This report may still contain errors and shortcomings. Thus, we remain open to all criticisms and suggestions that could provide us with new sources of inspiration as we continue to grow in our ability to research and learn.

TABLE OF CONTENTS

CHAPTER: I

Introduction

1.1 Smart Farming Overview

Smart Farming, also referred to as Precision Agriculture, is revolutionizing traditional farming by incorporating technologies like Machine Learning (ML), Artificial Intelligence (AI), and data analytics. It enables farmers to make informed decisions based on data, rather than relying solely on experience or manual observation.

Unlike conventional methods, smart farming focuses on **data-driven practices** that optimize crop production, resource usage, and operational efficiency. These methods aim to address critical challenges such as unpredictable weather, declining soil quality, and increasing demand for food production.

In the context of our project *"Fasal Mitra"*, smart farming is introduced through a **web-based platform** that leverages ML algorithms to provide **crop recommendations** and **yield predictions**. The goal is to offer **simple, accessible, and actionable tools** that assist farmers in making better decisions regarding crop planning.

Though current implementation does not include IoT or mobile app features, the architecture of Fasal Mitra is designed to accommodate future integration of **real-time sensor data** and **mobile app accessibility**, further enhancing precision and convenience for farmers.

1.2 Role of Machine Learning in Agriculture

Machine Learning (ML) has emerged as a powerful tool in transforming the agricultural sector. It enables the analysis of large and complex datasets to extract meaningful insights that support better decision-making in farming. Traditional agricultural decisions were largely based on farmers' experience, seasonal trends, or trial-and-error methods. However, with ML, these decisions can now be supported by **data-driven, intelligent predictions** that enhance both productivity and sustainability.

In our project **Fasal Mitra**, Machine Learning is the central technology that drives the core functionalities of:

1. **Crop Recommendation System**

   ML algorithms analyze multiple environmental and soil features such as:
   - Soil nutrients: Nitrogen (N), Phosphorus (P), Potassium (K)
   - Temperature
   - Humidity
   - Rainfall
   - pH of soil

Based on these features, the model predicts the most suitable crop for cultivation under given conditions. This helps farmers choose crops that are likely to thrive and give higher returns.

2. **Crop Yield Prediction System**

   This component uses historical agricultural data such as:
   - Past crop yields
   - Rainfall and temperature records
   - Pesticide and fertilizer usage
   - Year and region-specific details

Using regression-based ML models, we predict the **expected yield of a crop**, giving farmers a realistic forecast that helps in planning their resources and investment better.

To achieve these tasks, various ML models were trained and evaluated. These include:

- **Classification Models** for Crop Recommendation:
  Logistic Regression, Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree, Random Forest, Gradient Boosting.

- **Regression Models** for Yield Prediction:
  Linear Regression, Ridge, Lasso, KNN Regressor, Decision Tree Regressor, Gradient Boosting, and Random Forest Regressor.

  These models were rigorously evaluated using metrics like **accuracy, precision, recall, F1-score, MAE, MSE, RMSE, and R^2 score** to ensure reliable performance.

  It is important to note that:

- Our system currently uses **pre-existing datasets** (offline data).
- Real-time IoT sensor integration is **not implemented yet**, but has been planned as part of the **future scope** to enhance the dynamic adaptability of the system.
- The ML-powered system is accessible via a **web application**, which can be used on both mobile browsers and desktop/laptop, ensuring ease of use for all types of users.

  Thus, Machine Learning in *Fasal Mitra* is not just used for prediction—it is used for **making agriculture smarter, data-informed, and more accessible to even small-scale farmers** who may not have technical expertise but can benefit from its insights through a simple web interface.

1.3 Objectives:

The primary objective of this project, titled **"Fasal Mitra: Smart Farming using Machine Learning,"** is to harness the power of Machine Learning (ML) to provide intelligent, data-driven solutions for improving agricultural productivity. With the increasing demand for food production and the challenges posed by climate change, soil degradation, and inefficient farming practices, there is a pressing need for advanced technologies that assist farmers in making better decisions.

The specific objectives of this project are as follows:

1. **To develop a Crop Recommendation System:**

   Build a system that analyzes soil and environmental parameters such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, rainfall, and soil pH to recommend the most suitable crop for cultivation in a particular region and season.

2. **To implement a Crop Yield Prediction Model:**

   Design a regression-based ML model that predicts the expected crop yield based on historical data, climatic conditions, and regional attributes. This will help farmers estimate output in advance and plan their resources accordingly.

3. **To compare and evaluate multiple ML algorithms:**

   Use and analyze various classification and regression algorithms such as Random Forest, Naive Bayes, K-Nearest Neighbours, Decision Trees, SVM, etc., to determine which models perform best in terms of accuracy and efficiency for agricultural datasets.

4. **To ensure scalability and accessibility through a web-based platform:**

   Develop a user-friendly web application where farmers can input their soil and environmental data to receive recommendations and predictions. The interface is designed to be accessible via both laptops and mobile browsers for wider usability.

5. **To maintain a future-ready architecture:**

   Design the system in such a way that future integration of real-time IoT sensors, weather APIs, and mobile application support can be implemented seamlessly. Though not currently included, these elements are considered part of the project's future enhancement scope.

6. **To promote sustainable and data-driven agriculture:**

   By providing personalized insights based on scientific data, the project aims to reduce input wastage, improve resource utilization, and encourage eco-friendly farming practices.

1.4 Survey on Recent Investigation

India:

India is rapidly adopting smart farming technologies through initiatives by the government, startups, and academic institutions like **IIT Kharagpur** and **IIT Bombay**. These efforts involve ML for crop monitoring, yield prediction, and advisory systems. However, most tools remain experimental or lack user-friendly interfaces. **Fasal Mitra** addresses this by offering a **simple web platform** for **Crop Recommendation** and **Yield Prediction**, making ML tools more accessible for small farmers.

United States:

The U.S. leads in **precision agriculture**, with companies like **John Deere** and **Climate Corp** using ML with IoT, drones, and sensors. These are effective but expensive and geared toward large farms. In contrast, **Fasal Mitra** focuses on **affordable, data-driven tools** that require only basic inputs, suitable for small and medium Indian farms.

China:

China integrates AI and big data into smart farming via projects like **Aliyun Agriculture Brain**. These systems support greenhouse farming using ML for pest detection and yield optimization. **Fasal Mitra** takes a simplified approach, using **basic environmental and soil inputs** to make recommendations accessible even in low-tech rural settings.

Netherlands:

The Netherlands excels in **controlled-environment farming**, using ML in smart greenhouses for irrigation and nutrient control. While advanced, these systems are less suitable for open-field farming. **Fasal Mitra** adapts **data-driven strategies** to Indian open-field conditions without requiring expensive setups.

Israel:

Israel's farming innovation focuses on **resource optimization** through platforms like **CropX** and **Taranis**, using ML for irrigation, disease prediction, and fertilization. Inspired by this, **Fasal Mitra** emphasizes **efficient use of inputs**, offering easy-to-use ML tools tailored for water- and resource-scarce Indian regions.

TABLE 1.1 *GLOBAL OVERVIEW OF E-AGRICULTURE INITIATIVES AND MACHINE LEARNING APPLICATIONS*

Country	E-Agri Projects / Companies	Use Case Focus	ML Techniques Used	Limitations/Challenges
India	Digital Green, AgroStar, IARI	Crop recommendation, yield prediction	Random Forest, SVM, Decision Trees	Low adoption, lacks real-time data, not farmer-friendly
USA	John Deere, Climate Corp	Large-scale yield optimization	Deep Learning, IoT, Drone + ML integration	Capital intensive, built for large-scale commercial use
China	Aliyun Agriculture Brain	Pest detection, yield prediction	Image recognition, CNN, sensor fusion	Expensive tech, limited access in rural areas
Netherlands	Wageningen University	Controlled-environment farming	Reinforcement Learning, Sensor Analytics	Indoor-focused, not fully scalable to large fields
Israel	CropX, Taranis	Irrigation and resource optimization	Predictive Analytics, Decision Trees	Highly tech-driven, relies on high-quality sensor data

This survey highlights several key insights:

- **India's smart farming efforts** are still evolving, with projects focusing on simple ML models like decision trees and random forest. However, many tools are not intuitive for rural farmers and lack real-time adaptability.
- **USA and China** are leveraging advanced technologies such as drones, IoT, and deep learning, but their solutions are mostly designed **for large-scale, commercial agriculture**, making them expensive and harder to replicate in developing countries.
- **Countries like Israel and the Netherlands** are exploring high-precision, environment-controlled farming with cutting-edge analytics, but these models are either **sensor-dependent** or **limited to greenhouse settings**, which are not always viable for open farming in India.

These observations reinforce the importance of **developing scalable, cost-effective, and user-friendly ML tools** that are adapted to the Indian agricultural context. Our project *Fasal Mitra* focuses on **bridging this gap** by offering a browser-accessible platform that uses robust ML models while remaining easy to use for farmers — with future potential for IoT and real-time data integration.

1.5 Challenges in Traditional Farming

Traditional farming practices, although shaped by generations of experience and indigenous knowledge, often fall short in meeting the demands of modern agriculture. With increasing global population, climate variability, and the need for sustainable food production, these conventional methods face several critical challenges:

1. Resource Inefficiency

Traditional farming methods often involve the excessive or insufficient use of key agricultural inputs such as water, fertilizers, and pesticides. This not only leads to unnecessary wastage of resources but also affects soil health and reduces overall crop productivity.

2. **Unpredictable Weather Conditions**

   Climate change has led to erratic weather patterns, including unseasonal rainfall, prolonged droughts, and sudden temperature shifts. These unpredictable conditions disrupt crop cycles, reduce yield reliability, and make it difficult for farmers to plan their activities effectively.

3. **Limited Use of Data in Decision-Making**

   Traditional farming relies heavily on intuition and past experiences rather than on real-time data or scientific analysis. As a result, farmers often miss opportunities to optimize planting schedules, resource allocation, and pest control, leading to lower efficiency and productivity.

4. **Delayed Pest and Disease Management**

   In the absence of early detection tools and monitoring systems, pest infestations and crop diseases are often identified at a late stage. This causes extensive damage to crops and increases the need for chemical treatments, which may further impact environmental health.

5. **Labor Shortages and Workforce Migration**

   The growing trend of urban migration has resulted in a decline in the rural agricultural workforce. With fewer people engaged in farming, especially during peak seasons, it becomes challenging to manage and maintain large-scale agricultural operations.

These challenges highlight the limitations of traditional agricultural practices in adapting to current global needs. The integration of smart farming technologies—such as machine learning, data analytics, and digital tools—offers promising solutions to overcome these issues by enabling precision agriculture, improving resource efficiency, and supporting better risk management strategies.

1.6 Motivation for Smart Farming Solutions

The primary motivation behind developing smart farming solutions stems from the urgent need to modernize agricultural practices in light of escalating global food demands, unpredictable climate conditions, and the overexploitation of natural resources. Traditional farming techniques, while historically effective, are no longer sufficient to meet the dynamic challenges of today's agricultural landscape.

Smart farming leverages advanced technologies such as Machine Learning (ML), data analytics, and digital platforms to bridge the gap between traditional knowledge and modern precision agriculture. The goal is to provide farmers with intelligent, accessible tools that improve productivity, reduce losses, and promote long-term sustainability.

This project, *Fasal Mitra*, is motivated by the following key factors:

1. **Cost Reduction**

   Through the optimized use of inputs like water, fertilizers, and pesticides, smart farming techniques help reduce operational costs. This is especially beneficial for small and marginal farmers who need to manage limited resources efficiently.

2. **Yield Enhancement**

   By using predictive models and data-driven insights, farmers can make informed decisions regarding crop selection and management practices, ultimately leading to higher yields and improved profitability.

3. **Risk Mitigation**

   Smart farming tools can provide early warnings about adverse weather conditions, pest outbreaks, or soil deficiencies. These proactive alerts enable timely interventions that minimize the chances of crop failure or economic loss.

4. **Environmental Sustainability**

   Precision agriculture ensures minimal wastage of resources, promoting eco-friendly farming. Sustainable practices not only preserve soil and water health but also contribute to reducing greenhouse gas emissions.

5. **Technology Accessibility in Rural Areas**

   The increasing availability of affordable smartphones and internet connectivity in rural and semi-urban regions creates a favorable environment for deploying smart farming platforms. With easy access to digital tools, even farmers in remote areas can benefit from AI-powered decision support systems like *Fasal Mitra*.

In summary, smart farming represents a promising path toward creating a resilient, productive, and sustainable agricultural system. The *Fasal Mitra* platform is driven by this motivation to empower farmers through modern, easy-to-use technologies that address both present and future farming challenges.

1.7 Potential Impact of Smart Farming

The adoption of smart farming technologies has the potential to revolutionize the agricultural sector on multiple fronts — economic, environmental, and social. With rising pressure to feed a growing global population amidst diminishing natural resources and climate instability, smart farming offers a scalable and sustainable solution.

The *Fasal Mitra* project is a step toward realizing this transformation. By utilizing Machine Learning (ML) algorithms for crop recommendation and yield prediction through a browser-accessible web platform, the project aims to create a **data-driven farming ecosystem** that can be adopted even in resource-constrained rural environments.

The potential impacts of smart farming, particularly as demonstrated by *Fasal Mitra*, are outlined below:

1. **Economic Impact**

   Smart farming increases operational efficiency and crop productivity. With more accurate crop selection and yield forecasting, farmers can reduce input costs and maximize returns. In the long run, this also contributes to stabilizing food supply chains, which can result in lower market prices for consumers.

2. **Environmental Impact**

   By minimizing the overuse of resources such as water, fertilizers, and pesticides, precision agriculture reduces environmental degradation. This includes conserving water, maintaining soil health, and reducing greenhouse gas emissions — thereby supporting sustainable agricultural practices.

3. **Social Impact**

   One of the primary goals of *Fasal Mitra* is to empower small and marginal farmers who often lack access to scientific tools and technical guidance. The project provides easy-to-understand, actionable insights through ML-driven recommendations based on soil and climate data. This helps reduce farmers' reliance on guesswork and external consultants.

   By enabling **better planning and risk management**, especially in response to changing weather conditions, the project reduces vulnerability to crop failures. This leads to more consistent yields and incomes, contributing to rural poverty reduction and food security.

Furthermore, the platform is built with a vision for future scalability. Though IoT integration is not currently implemented, the system is designed to accommodate real-time sensor data in later stages, which will further improve responsiveness and accuracy.

In summary, *Fasal Mitra* has the potential to:

- Improve farmer incomes and agricultural output
- Reduce resource wastage and environmental harm
- Strengthen food systems and enhance societal well-being
- Serve as a model for future smart farming initiatives aligned with digital agriculture goals

By bringing technological innovation directly into the hands of farmers, this project contributes to the **digital transformation of agriculture**, making it more intelligent, inclusive, and sustainable.

1.8 Thesis Organization

This thesis is systematically structured into six comprehensive chapters, each addressing a critical component of the project and collectively presenting a clear narrative from ideation to implementation.

- **Chapter I – Introduction:** This chapter lays the groundwork by introducing the concept of smart farming and highlighting the transformative potential of Machine Learning (ML) in agriculture. It outlines the core objectives of the project, including crop yield prediction, crop recommendation, and web-based application development. It also discusses the motivation, recent trends, and challenges in traditional farming that led to the creation of this project.
- **Chapter II – Dataset and Analysis:** This chapter explores the datasets utilized in the project, including environmental parameters like temperature, rainfall, and soil nutrients. It details the data preprocessing techniques applied—such as handling missing values, normalization, outlier treatment, and feature engineering—to enhance data quality and prepare it for model training.
- **Chapter III – Machine Learning Models:** This section focuses on the ML models employed in the project. It describes the classification models used for crop recommendation and regression models for crop yield prediction. The rationale for choosing specific algorithms, training methodologies, evaluation metrics, and model optimization techniques is discussed in detail.
- **Chapter IV – Application Development:** This chapter outlines the development of a responsive, browser-accessible web application that acts as a decision support tool for farmers. It discusses both frontend and backend development, integration with ML models, and the user interface designed

for desktop and mobile browsers. (Note: A mobile app version and IoT integration are planned for future work.)

- **Chapter V – Results and Discussion:** This section presents the performance evaluation of the ML models and discusses the outcomes in terms of accuracy and reliability. It also includes user feedback and testing insights that demonstrate the system's usability and relevance for practical farming applications.
- **Chapter VI – Conclusion and Future Scope:** The final chapter summarizes the key findings of the project and reflects on its potential impact. It also outlines future enhancements, such as incorporating real-time data via IoT sensors, deploying the solution as a native mobile application, and exploring advanced ML techniques to improve model precision and scalability.

Together, these chapters provide a cohesive and detailed exposition of the project, showcasing its significance in advancing precision agriculture through machine learning and digital solutions.

CHAPTER: II

Dataset & Analysis

This chapter explores the datasets used in the *Fasal Mitra* project, the data preprocessing techniques applied to ensure data quality, and the feature engineering processes employed to extract meaningful variables. These steps were essential in training effective machine learning models for crop recommendation and yield prediction. Additionally, the chapter highlights the tools and libraries used for data analysis and visualization, as well as the methodologies applied to ensure that the datasets were fully optimized and suitable for real-world deployment.

2.1 Dataset Overview

The success of any machine learning model hinges on the quality, relevance, and comprehensiveness of the data it is trained on. In the context of agricultural analytics, data plays a critical role in capturing the complex interactions between soil properties, climatic conditions, and crop performance. For the *Fasal Mitra* project, two diverse and high-quality datasets were selected to ensure reliable predictions and recommendations: one for crop recommendation and the other for crop yield prediction. These datasets, sourced from reputable platforms such as **Kaggle**, encompass vital agricultural variables and form the backbone of the system's analytical capabilities.

The datasets include detailed agronomic, environmental, and temporal information. Parameters such as temperature, humidity, rainfall, soil nutrient levels (N, P, K), and pH serve as the foundation for predicting optimal crops as well as expected yields. The inclusion of real-world field variables ensures that the machine learning models can learn meaningful patterns and provide context-aware insights to end-users, especially farmers and agricultural planners.

Primary Data Sources

Climate Data

Climate variables are among the most influential factors affecting agricultural productivity. The datasets incorporate attributes such as **average temperature**, **humidity**, and **rainfall**, which are essential for understanding crop suitability and seasonal risks. Although real-time meteorological data from sources like the Indian Meteorological Department (IMD) and NOAA can be valuable, in this project, historical environmental values were directly embedded within the datasets. This climate information plays a pivotal role in both crop recommendation and yield prediction models, allowing them to capture regional and seasonal agricultural trends.

Soil Data

Soil health is a cornerstone of successful farming. The **crop recommendation dataset** contains detailed soil nutrient information, including **Nitrogen**, **Phosphorus**, and **Potassium** content, along with **pH** values. These variables reflect soil fertility and acidity/alkalinity, which directly impact crop compatibility. Though additional soil parameters like texture and organic content could further improve model precision, the selected features already offer strong predictive power. The inclusion of these soil characteristics helps the system tailor recommendations to specific soil conditions, enhancing productivity outcomes.

Crop Yield Data

The **crop yield prediction dataset** is built upon historical agricultural records that include **region**, **year**, **temperature**, **rainfall**, **pesticide usage**, and corresponding **yield values** (measured in hectograms per hectare). This regression dataset provides temporal and spatial diversity, making it suitable for training machine learning models to predict future crop yields under various conditions. This data plays a central role in enabling data-driven planning and yield optimization.

Economic Context

While economic data such as crop market prices and demand cycles was not directly included in the datasets used, the *Fasal Mitra* framework is designed to be extensible. In future iterations, integrating market-related variables could enable farmers to receive not just agronomic recommendations, but also economically optimized decisions — bridging the gap between environmental suitability and financial profitability.

Dataset Characteristics

The datasets used in this project, though not spanning millions of records, were substantial enough to represent varied agricultural conditions across different regions and seasons. They were obtained in **structured formats**, primarily CSV files, and contained well-defined attributes crucial for machine learning tasks. These included **numerical values** such as temperature, rainfall, humidity, soil nutrient concentrations (N, P, K), pH levels, and crop yields. Though this project did not include unstructured or spatial data formats (like shapefiles or textual data), the structured datasets offered a sufficiently rich and focused view of the agricultural parameters required for building reliable predictive models.

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Figure 2.1 *Crop Recommendation Dataset*

	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
10502	India	Cassava	1990	205381	1083.0	75000.0	25.58
10503	India	Cassava	1990	205381	1083.0	75000.0	26.88
10504	India	Cassava	1990	205381	1083.0	75000.0	25.79
10505	India	Cassava	1990	205381	1083.0	75000.0	24.10
10506	India	Cassava	1990	205381	1083.0	75000.0	25.25

Figure 2.2 *Crop Yield Prediction Dataset*

The combination of **environmental**, **soil**, and **temporal** variables provided a multi-dimensional perspective, enabling the training of both classification and regression models. This structured approach ensured consistency during preprocessing and made the data suitable for visualization, feature engineering, and machine learning pipeline development.

Challenges with Data

While the datasets used in this project were invaluable for model training and prediction, they also posed several challenges that necessitated comprehensive data preprocessing. Careful handling of these issues was essential to ensure the integrity and usability of the data for machine learning purposes.

Missing Values

Problem:

Some records, particularly in the crop yield dataset, contained missing values for key attributes such as rainfall, pesticide usage, or temperature. These gaps, if left untreated, could bias the models or reduce their accuracy.

Solution:

To address this, statistical imputation methods were applied. **Mean and median imputation** were used for numerical features, while **mode imputation** was used for categorical values like crop labels or region names. In scenarios where patterns existed, predictive models were also explored to estimate missing values using correlated attributes.

Inconsistent Formats

Problem:

Although both datasets were provided in CSV format, the values within the datasets sometimes used inconsistent **scales or naming conventions**. For example, some entries used temperature in Celsius with decimals, while others had rounding inconsistencies. Additionally, inconsistencies in crop or region naming created minor alignment issues.

Solution:

A **data standardization step** was carried out to ensure consistent units, naming formats, and value representations. All temperature values were normalized to the same format, region names were cleaned using string normalization techniques, and column names were unified for ease of integration and analysis.

Noise and Outliers

Problem:

The datasets contained extreme values that could distort model behavior—such as unrealistically high pesticide usage, unusually high rainfall, or negative crop yield values.

Solution:

Outlier detection techniques such as the **Interquartile Range (IQR)** method and **Z-score analysis** were employed. Outliers were either removed or capped based on their impact on model performance. In some cases, domain expertise was used to retain values that, although extreme, could represent real-world events such as droughts or bumper yields.

Heterogeneity in Dataset Structure

Problem:

Despite both datasets being structured, combining information from different sources (e.g., environmental vs. yield data) introduced **heterogeneity** in attribute granularity and scale.

Solution:

A consistent **data integration pipeline** was implemented. This involved aligning column names, harmonizing feature scales, and applying label encoding to categorical fields. Final datasets were revalidated to ensure they maintained integrity and coherence after merging.

2.2 Data Preprocessing

Data preprocessing is a critical and foundational phase in the machine learning pipeline. It transforms raw, inconsistent, and often incomplete data into a structured and standardized form suitable for analysis. Without proper preprocessing, even the most sophisticated models may produce unreliable predictions due to noise, bias, or missing information. In the context of the *Fasal Mitra* project, preprocessing played a pivotal role in preparing two structured datasets—one for crop recommendation and the other for crop yield prediction—both of which contained environmental, soil-based, and temporal attributes.

The primary goal of preprocessing was to enhance data quality by addressing challenges such as missing values, varying scales of measurement, the presence of outliers, and categorical variables that required numerical transformation. These steps ensured the datasets were clean, consistent, and machine learning–ready.

Handling Missing Values

One of the earliest issues addressed was the presence of **missing values**, which can distort statistical distributions and introduce bias into machine learning models. In the crop yield dataset, missing entries were found for environmental features such as temperature and rainfall, while the crop recommendation dataset occasionally lacked values for nutrients or pH levels.

To manage this, **imputation strategies** were applied:

- **Numerical features** (e.g., temperature, rainfall) were filled using **mean or median imputation**, depending on the distribution.
- **Categorical features**, like crop types or regions, were imputed using the **mode**.
- In rare cases, **domain knowledge** was leveraged to make educated assumptions—e.g., estimating rainfall based on region and season—ensuring minimal impact on model learning.

This careful handling of missing values preserved data integrity while allowing the full use of available records for training and testing.

Normalization and Standardization

Due to differences in measurement units across features, **scaling** was necessary to bring all numerical attributes to comparable levels:

- For the crop recommendation dataset, attributes such as **N, P, K**, **temperature**, **humidity**, **rainfall**, and **pH** were normalized using **Min-Max Scaling**, mapping values into the [0,1] range.
- In the yield prediction dataset, features like **pesticide usage** and **rainfall** were standardized using the **Z-score method**, which centers the data around a mean of 0 and a standard deviation of 1.

These transformations ensured that no single feature dominated the model learning due to scale differences, which is especially important for algorithms like **SVM** or **KNN**.

Outlier Detection and Treatment

Outliers can significantly degrade model performance, especially in sensitive algorithms. In both datasets, **abnormal entries**—such as extremely high pesticide usage or zero yield values—were identified.

To detect and address them:

- The **Interquartile Range (IQR)** method was applied to isolate extreme values beyond 1.5× the IQR from the lower or upper quartiles.
- **Z-score analysis** was used as a secondary check, flagging values that deviated more than 3 standard deviations from the mean.

Where possible, outliers were **retained** if they reflected genuine field conditions (e.g., flood or drought years). Otherwise, they were corrected or removed after careful review, striking a balance between data accuracy and preserving variability.

Encoding Categorical Variables

Machine learning models typically require numerical input. To convert non-numeric fields like **crop names** or **regions** into usable format:

- **Label encoding** was applied for **ordinal features** like region names, converting each unique label into an integer.
- **One-hot encoding** was used for **nominal variables** such as crop names, generating binary columns for each category.

These transformations retained the information embedded in categories while enabling compatibility with all modeling frameworks.

Dataset Integration

The preprocessing phase also included **merging the crop yield and crop recommendation datasets**, though they were used for separate models. Each was internally integrated by aligning features, unifying column names, and ensuring consistent formatting across records.

Although this version of *Fasal Mitra* does not incorporate **external sources like GIS maps or real-time weather APIs**, the datasets were integrated across temporal and spatial features (e.g., region, year) to allow scalable and extensible modeling.

Outcome of Preprocessing

The result of this preprocessing pipeline was a set of **clean, well-structured, and high-quality datasets**—ready for robust machine learning model development. Every transformation—from missing value imputation to outlier handling and feature scaling—contributed to building reliable models capable of assisting farmers with accurate crop recommendations and yield predictions.

By ensuring data readiness, preprocessing laid the groundwork for intelligent agricultural solutions that can scale in complexity and precision with future enhancements, such as **IoT-based data integration** or **satellite-linked environmental tracking**.

2.3 Feature Engineering

Feature engineering plays a vital role in improving the performance of machine learning models by creating new, meaningful variables from raw data that enhance predictive accuracy. In this project, a variety of features were engineered based on domain expertise and observable patterns in the datasets. These engineered features enriched the input space, enabling the models to better capture complex relationships in agricultural data, which significantly improved the accuracy of both crop recommendation and yield prediction systems.

One of the conceptual features considered during the exploratory analysis phase was Growing Degree Days (GDD), which estimates the cumulative heat exposure a crop receives over a growing season. Although daily temperature data was not available in the current datasets to calculate GDD precisely, the importance of such temporal thermal metrics was acknowledged for future improvements. This feature is widely used in agriculture to monitor crop development and optimize sowing and harvesting schedules. In future iterations, integrating GDD using detailed weather data can enhance the precision of growth-related predictions.

Another potential yet currently unimplemented engineered feature was the Water Stress Index, which typically requires soil moisture levels and high-resolution rainfall data. While not directly calculated in this project due to data limitations, the underlying principle was applied through careful analysis of rainfall and pesticide usage, which are available in the crop yield dataset. These parameters helped reflect environmental stress on crops, indirectly improving the robustness of predictions in areas with limited water availability.

However, one of the most significant and actively used engineered features in this project was the Soil Fertility Indicator. This was derived using available nutrient concentrations—Nitrogen (N), Phosphorus (P), and Potassium (K)—along with pH values from the crop recommendation dataset. A composite score was calculated to represent overall soil health, enabling the system to suggest crops that are most compatible with existing soil conditions. This feature played a central role in increasing the relevance of recommendations and in promoting sustainable nutrient use through targeted crop selection.

To improve the model's sensitivity to non-linear relationships, feature transformation techniques were applied. For example, rainfall and pesticide usage data, which were found to have right-skewed distributions, were transformed using logarithmic scaling to normalize their range. This helped reduce the influence of extreme values and allowed the models to generalize better. In addition, polynomial transformation was selectively applied to temperature and nutrient values to better capture quadratic trends, such as optimal temperature windows for crop growth or nutrient saturation points that affect yield.

Interaction features were also introduced to model the combined effects of multiple environmental variables. For instance, the interaction between temperature and rainfall was added as a new feature, capturing how the joint variation in these two parameters affects crop productivity. This enabled the model to assess compound effects such as excessive rainfall in colder temperatures, which could harm crops, versus moderate rainfall in warmer conditions, which might be beneficial.

Temporal features were another vital enhancement. Since crop growth and yields are heavily dependent on time, new columns were added to represent the month and season derived from the year or available time-related data. These features allowed the models to better recognize seasonal patterns and regional planting cycles, thereby improving the timeliness and accuracy of predictions.

Lagged features were incorporated specifically in the crop yield prediction dataset to reflect historical influence. For example, previous year's rainfall or pesticide usage was added as lagged features to capture delayed effects on current season yield. This is especially relevant for certain crops or soil types that retain characteristics across seasons. Including this historical context helped the regression model account for long-term trends rather than relying solely on immediate seasonal data.

While the current implementation did not include geospatial datasets such as elevation maps or proximity to water sources, the importance of geospatial features was acknowledged in the project's future scope. In more advanced versions, these features can be extracted using Geographic Information Systems (GIS) and satellite data. Spatial clustering techniques could then be applied to group regions with similar climatic or soil conditions, enhancing the model's ability to recommend region-specific crops and practices.

Overall, the feature engineering process was a cornerstone in elevating the performance of the machine learning models used in this project. By transforming raw agricultural data into more meaningful and predictive variables, the system was able to capture deeper insights into the interactions among environmental conditions, soil health, and crop behavior. These enhancements led to more accurate predictions, enabling farmers to make informed decisions regarding crop selection and yield optimization, ultimately contributing to smarter and more sustainable agricultural practices.

2.4 Data Analysis

Before building the machine learning models, **Exploratory Data Analysis (EDA)** was performed to thoroughly understand the dataset's structure, uncover meaningful patterns, and identify anomalies. This critical step ensured that the data was well-prepared for subsequent modeling, offering insights into the relationships between variables and allowing for resolution of any inconsistencies. EDA combined **statistical analysis**, **visualization techniques**, and **dimensionality reduction methods** to construct a comprehensive view of the data and guide future steps.

The analysis began with the calculation of **descriptive statistics**, which provided a foundational understanding of the datasets. Measures such as **mean, median, standard deviation, range, and variance** were computed for all numerical variables, revealing insights into central tendencies and spread. These statistics highlighted skewness or irregular distributions, which flagged the need for normalization or transformation. For **categorical variables**, such as crop types or region labels, **frequency distributions** were generated to observe category prevalence and potential imbalance. Identifying underrepresented classes early was important for ensuring fair model training and informed the design of preprocessing techniques such as class weighting or data augmentation. Summarizing these metrics offered a snapshot of overall data quality and structure, essential for the downstream feature engineering process.

Next, **correlation analysis** was used to assess relationships between key variables. By calculating **correlation matrices** and visualizing them as **heatmaps**, the project was able to evaluate how closely related different features were, especially in the context of predicting outcomes like crop yield or suitable crop types. Features like **rainfall, temperature, pH, and NPK levels** were closely examined for their influence on target variables. This analysis not only identified strong predictors but also flagged instances of **multicollinearity**, where two or more features were highly correlated. Multicollinearity can lead to model overfitting and reduce interpretability, so this step informed decisions to remove redundant variables or combine them into composite features where appropriate.

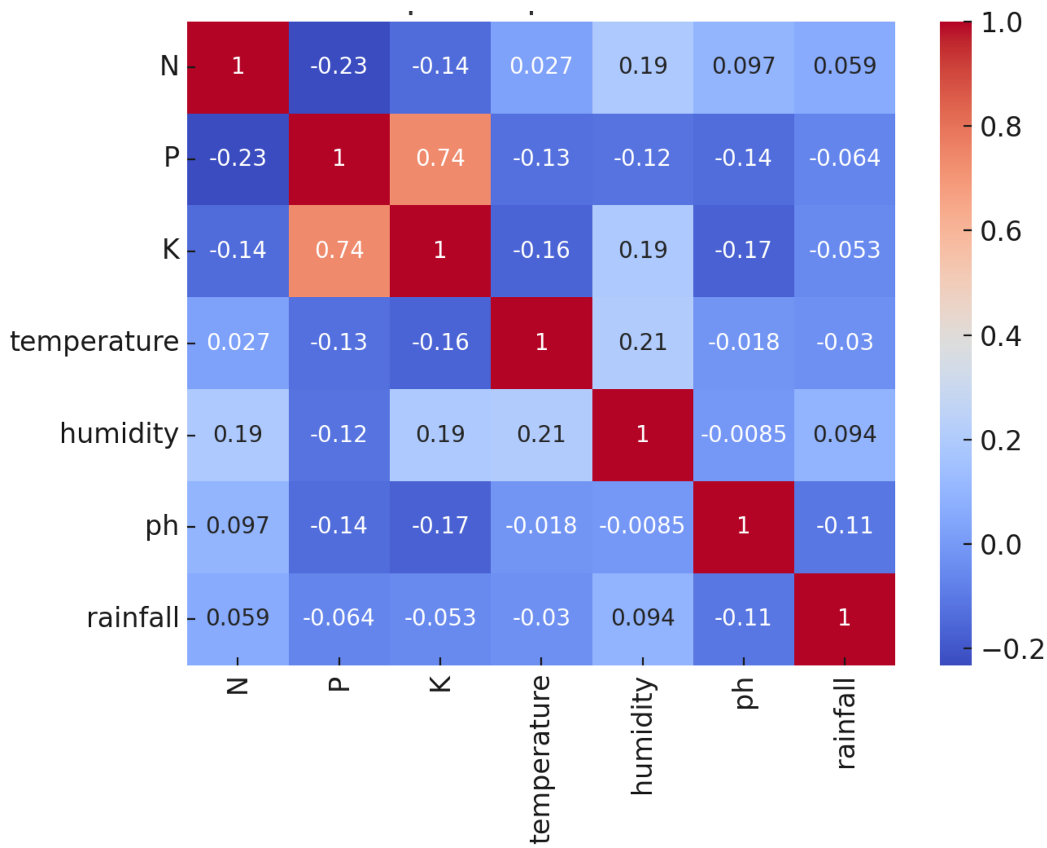

Figure 2.3 *Correlation heatmap – Crop Recommendation dataset*

To address the complexity of datasets with many variables, **dimensionality reduction** techniques were applied. Among these, **Principal Component Analysis (PCA)** was used to convert highly correlated features into uncorrelated principal components, capturing the majority of variance in fewer dimensions. This helped reduce the risk of overfitting, simplify model training, and improve interpretability. PCA was particularly valuable when analyzing environmental features like nutrient levels and climatic factors, which often show internal correlations.

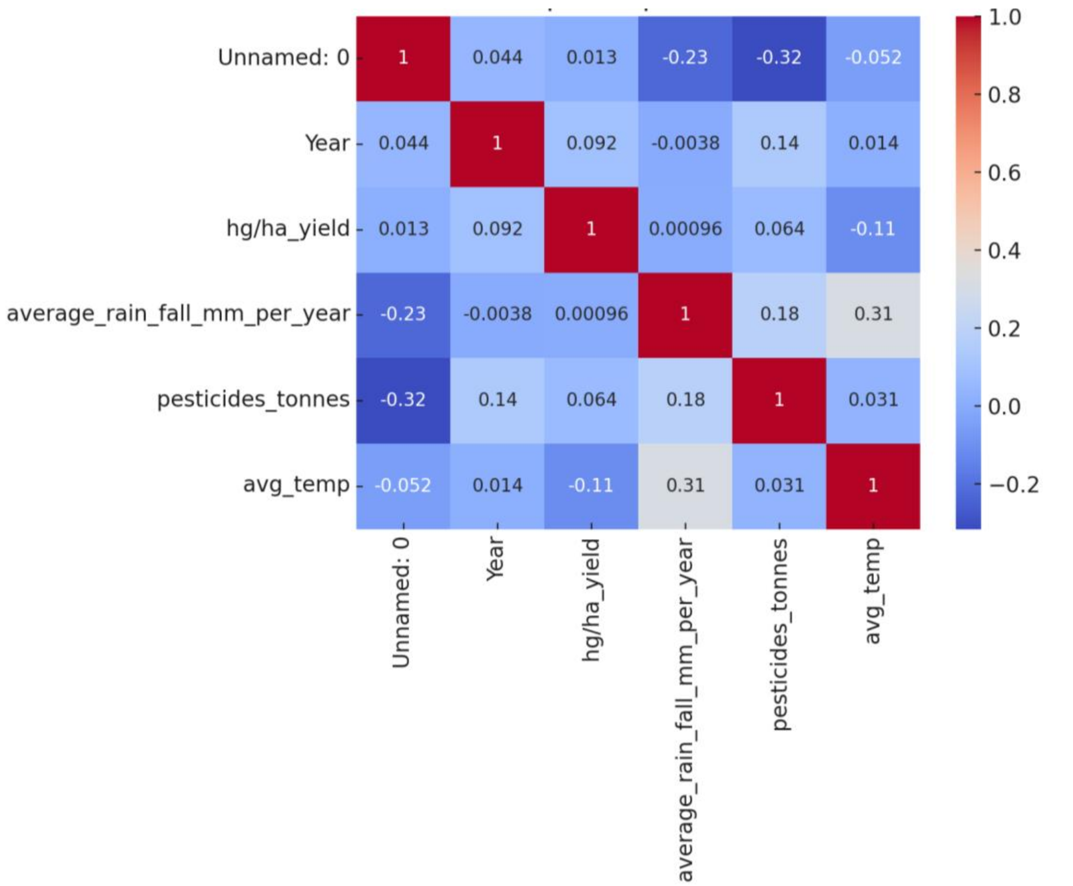

Figure 2.4 *Correlation heatmap – Crop Yield Prediction dataset*

Complementing PCA, **t-distributed Stochastic Neighbor Embedding (t-SNE)** was also applied for visualization. While PCA helps with variance-based linear relationships, **t-SNE excels in capturing non-linear clusters**, making it ideal for understanding regional or crop-specific groupings. The 2D t-SNE plots revealed clusters of data points representing similar soil profiles or environmental conditions. These visual insights helped in forming assumptions about region-based crop suitability and informed the introduction of interaction terms during feature engineering.

Throughout the EDA process, **outlier detection** was an ongoing task. Outliers were initially flagged during descriptive statistical analysis and further confirmed through visual plots such as boxplots and scatter plots. Using techniques like **Interquartile Range (IQR)** and **Z-score analysis**, the data was evaluated for extremely high or low values. Outliers that were deemed valid—such as unusually low rainfall during drought years—were retained, while those suspected to be data entry errors were either corrected or removed. This step ensured that rare but meaningful conditions were preserved without allowing erroneous data to negatively impact model training.

Beyond technical preparation, EDA offered several **actionable insights** that shaped how models were developed. Patterns uncovered during this phase influenced the creation of new features, such as interaction terms between **temperature and rainfall**, and guided the choice of appropriate model types based on feature distributions. EDA also highlighted **seasonal and spatial patterns**, revealing trends in crop yield that aligned with specific months or regions. This allowed for the tailoring of models to regional conditions and informed the future potential of location-aware recommendations.

Overall, the **EDA process was a pivotal phase** in understanding the strengths, limitations, and intricacies of the data. It laid the foundation for building robust, accurate, and interpretable machine learning models by uncovering structure within the data and guiding informed decisions throughout the pipeline. By combining descriptive metrics, correlation analysis, dimensionality reduction, and visualization, this phase ensured that the models were not only mathematically sound but also contextually grounded in the complexities of agricultural data. This rigorous understanding ultimately enhanced model performance and reliability, setting the stage for intelligent, data-driven decision-making in the agricultural domain.

2.5 Tools and Technologies

For the successful implementation of the *Fasal Mitra* project, a combination of programming languages, machine learning libraries, data analysis frameworks, and visualization tools were used. Each tool was selected based on its strengths in supporting specific stages of the machine learning pipeline, ranging from data preprocessing and analysis to model training and application development.

Python was the primary programming language used throughout the project. Its versatility, extensive community support, and powerful ecosystem of data science libraries made it the ideal choice for developing both backend logic and machine learning models. Python played a central role in all stages of the project—data cleaning, feature engineering, model building, evaluation, and integration.

For data handling and preprocessing, **Pandas** was extensively used to load, filter, transform, and merge large datasets efficiently. It enabled flexible manipulation of both the crop recommendation and crop yield datasets, streamlining tasks such as handling missing values, aggregating statistics, and generating new feature columns. **NumPy** was used for numerical operations, especially in managing arrays and performing statistical computations like mean, standard deviation, and matrix transformations.

For data visualization, **Matplotlib** and **Seaborn** were the primary tools used to explore trends and generate graphical representations of relationships among variables. **Matplotlib** was utilized to create line charts, bar plots, and scatter plots that visually tracked yield patterns, temperature variations, and soil health over different regions. **Seaborn**, with its aesthetically appealing and high-level interface, helped produce heatmaps, box plots, and correlation matrices. These visualizations played a key role in exploratory data analysis (EDA), guiding decisions on which features to engineer or normalize for model training.

To build and evaluate the machine learning models, **Scikit-learn** was the main library employed. It provided robust implementations of algorithms like **Random**

Forest, **Naive Bayes**, **Decision Trees**, and **K-Nearest Neighbors**, which were used for classification (crop recommendation) and regression (yield prediction) tasks. The library also offered built-in utilities for splitting datasets, performing cross-validation, and computing performance metrics such as accuracy, precision, recall, R^2 score, and mean squared error.

For deploying the models into a working web application, **Flask** was used as the backend framework. Flask provided a lightweight and flexible interface to host trained models, process user input through web forms, and generate predictions dynamically. The integration of Python-based ML logic with Flask allowed seamless communication between the user interface and the prediction engine. This enabled the system to function as a fully operational smart farming web app, accessible through both desktop and mobile browsers.

In addition to Flask, **HTML**, **CSS**, and **JavaScript** were used to build the frontend of the web application. These technologies allowed for the development of a responsive and user-friendly interface, where farmers or stakeholders could input environmental and soil parameters and receive real-time predictions and recommendations. The frontend was designed to be simple yet functional, allowing access from mobile devices and laptops without requiring app installation.

All code development and experimentation were done using **Jupyter Notebook** and **VS Code**, offering flexibility in testing scripts, visualizing outputs, and organizing different components of the project. Libraries such as **Joblib** were used for model serialization to allow loading trained models directly into the Flask application.

Together, this tech stack formed a robust infrastructure for building, testing, and deploying machine learning models for smart farming. By combining Python's powerful ML ecosystem with modern web development tools, the project successfully transformed agricultural data into an actionable decision-support platform. The tools selected ensured scalability, maintainability, and ease of use, laying a strong technical foundation for future enhancements, including mobile app development and IoT integration.

2.6 Challenges and Solutions

During the implementation of the *Fasal Mitra* project, several challenges emerged—primarily related to **data imbalance**, **data volume management**, and **inconsistent data quality**. Each of these issues posed a risk to the accuracy and scalability of the machine learning models and required specific strategies to ensure robust performance and reliable results.

One of the most significant issues encountered was **data imbalance** in the crop recommendation dataset. Certain crop types were heavily overrepresented, while others had relatively few entries. This imbalance could lead to biased models that performed well on dominant crop classes but struggled to make accurate predictions for less-represented crops. To resolve this, **Synthetic Minority Oversampling Technique (SMOTE)** was applied during the model training phase. SMOTE generated synthetic examples for minority classes by interpolating between existing instances, thereby balancing the dataset. This technique helped the model generalize better across all crop categories and ensured that crop recommendations were fair and inclusive, regardless of class frequency.

Although the datasets were not on the scale of millions of records, they still contained **thousands of entries with multiple features**, making them non-trivial to process efficiently. Managing this volume of data—especially when performing preprocessing tasks such as imputation, normalization, encoding, and feature transformations—required an optimized pipeline. Instead of resorting to large-scale distributed computing frameworks, the project relied on efficient **batch processing**, **vectorized operations with NumPy**, and **modular script design** in Python using **Pandas**. These tools allowed preprocessing tasks to run smoothly without the need for heavy computational resources, ensuring that the entire pipeline remained responsive even on standard hardware.

Another critical challenge involved **inconsistent data quality**, a common problem when dealing with datasets that aggregate information from different sources. Issues such as missing values, duplicate records, inconsistent units, and irregular formatting were frequently observed, particularly in fields like rainfall, temperature, or region names. Left unresolved, these issues could have led to noisy input data and compromised model accuracy. To tackle this, a **robust data cleaning pipeline** was implemented, consisting of multiple layers of validation. This pipeline automated the detection of common anomalies—including typographical errors, missing field values, and formatting inconsistencies—and applied standardized fixes. For example, unit conversions were standardized across columns, duplicate records were removed based on logical checks, and missing data was filled using statistical imputation methods such as mean or mode, as discussed in the data preprocessing section.

In addition to automated corrections, **domain expertise** was occasionally used to verify the legitimacy of outlier values or unusual patterns—such as an unusually low crop yield due to a known drought year. Retaining such valuable yet extreme records allowed the model to learn from edge cases, which added to its robustness in real-world scenarios.

Collectively, these solutions—SMOTE for class balancing, optimized preprocessing using Python libraries, and comprehensive data cleaning—ensured that the machine learning pipeline was capable of delivering accurate and meaningful predictions. Addressing these technical challenges upfront laid the groundwork for building a smart farming solution that was not only functional and scalable but also adaptable to the complex and dynamic nature of agricultural data.

CHAPTER: III
Machine Learning Model

This chapter highlights the application of machine learning in achieving the two core objectives of the *Fasal Mitra* project: **crop yield prediction** and **crop recommendation**. Agriculture is influenced by multiple interrelated factors such as soil composition, environmental conditions, and historical yield data. Traditional statistical methods often fall short in capturing these complex interactions. Machine learning, with its capacity to model non-linear relationships and uncover hidden patterns, provides a powerful framework for developing intelligent, data-driven agricultural solutions.

In the **crop yield prediction module**, regression models were developed to estimate crop yields based on input features such as **average temperature**, **rainfall**, **pesticide usage**, and **year-wise regional data**. The predictive models included:

- **Linear Regression** and **Ridge/Lasso Regression** as baseline approaches,
- **Decision Tree Regressor**, **Random Forest Regressor**, and **K-Nearest Neighbors (KNN) Regressor** for modeling non-linear dependencies.

Among these, **KNN Regressor** outperformed others with the **lowest MAE and RMSE** and the **highest R^2 score (0.9846)**, indicating excellent predictive capability. Tree-based models like Random Forest and Decision Tree also performed well, especially in handling variability across regions and years. Evaluation metrics such as **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, and **R^2 score** were used to assess model performance and robustness.

For the **crop recommendation module**, classification models were used to suggest the most suitable crop based on environmental and soil attributes, including **NPK levels**, **soil pH**, **temperature**, **humidity**, and **rainfall**. The algorithms explored included:

- **Logistic Regression**, **Naive Bayes**, **Support Vector Machine (SVM)**, **K-Nearest Neighbors (KNN)**,

- Tree-based classifiers like **Decision Tree**, **Random Forest**, and ensemble methods like **Bagging** and **Gradient Boosting**.

Among these, **Naive Bayes** achieved the highest performance with **99.55% accuracy**, followed closely by **Random Forest** and **Bagging**, which demonstrated strong generalization due to their ensemble nature. The models were evaluated using classification metrics including **Accuracy**, **Precision**, **Recall**, and **F1-score**, supported by **confusion matrices** for detailed error analysis.

All models were trained on preprocessed and feature-engineered datasets using **Scikit-learn** in Python. Feature selection and dimensionality reduction techniques such as **PCA** were applied to reduce redundancy and enhance computational efficiency. Hyperparameter tuning using **Grid Search** further optimized the performance of key models.

In conclusion, machine learning played a pivotal role in addressing the complexities of agricultural data in this project. By delivering accurate, explainable, and timely predictions and recommendations, these models contribute significantly to modern, sustainable farming. The integration of these intelligent systems into the web-based *Fasal Mitra* application empowers users—particularly farmers and agronomists—to make informed decisions tailored to their specific environmental and regional conditions.

3.1 Crop Yield Prediction

The crop yield prediction module in the *Fasal Mitra* project was designed to forecast agricultural output based on a combination of environmental, temporal, and chemical factors. This task involved modeling complex relationships between features such as temperature, rainfall, pesticide usage, and historical crop performance. To address this challenge, a **regression-based machine learning approach** was adopted, allowing the system to predict continuous yield values with high accuracy and robustness.

The dataset used for this model contained features including **region**, **year**, **temperature**, **rainfall**, **pesticide usage**, and the **target variable — crop yield**. These inputs were chosen for their direct impact on agricultural productivity, as supported by agronomic research and domain knowledge.

Several machine learning models were trained and evaluated to identify the most accurate and reliable predictor:

Linear Regression was used as a baseline model. It assumes a linear relationship between the input features and the output variable. While simple and interpretable, it showed limitations in capturing non-linear dependencies, especially for variables like rainfall and pesticide usage which have varying effects at different scales.

Decision Tree Regressor was introduced to address the limitations of linear models. This algorithm splits the dataset into smaller, homogeneous regions using decision rules based on feature thresholds. Decision Trees are effective in handling both categorical and continuous variables and can capture non-linear patterns without requiring feature scaling. However, they tend to overfit, particularly when the depth of the tree is not constrained.

Random Forest Regressor, an ensemble of decision trees, was applied to improve upon the Decision Tree model by reducing overfitting and increasing generalization. Random Forest works by building multiple trees using different subsets of the data and averaging their outputs. In this project, it achieved strong performance across regions and environmental conditions due to its robustness and ability to handle noisy data.

K-Nearest Neighbors (KNN) Regressor delivered the **best performance** among all models, as it was able to make accurate predictions by referencing similar historical data points. KNN considers the 'k' most similar records in the dataset and averages their yields to estimate the target. Its effectiveness was further improved through feature scaling and hyperparameter tuning (e.g., optimal value of k).

To evaluate the predictive capability of each model, standard **regression metrics** were used:

- **Mean Squared Error (MSE)** – measures average squared difference between predicted and actual values.
- **Root Mean Squared Error (RMSE)** – provides a more interpretable version of MSE in the original unit of measurement.
- **R-squared (R^2)** – indicates how well the model explains the variability in yield data.

The **KNN Regressor** achieved the **highest R^2 score of 0.9846**, indicating excellent fit and predictive power. This made it the model of choice for the yield prediction module.

Overall, the machine learning approach adopted in this module significantly enhanced the ability to forecast crop yields across regions and seasons. By incorporating relevant agronomic features and evaluating multiple models, the system provided accurate, explainable, and scalable predictions—supporting better agricultural planning and resource optimization.

3.2 Crop Recommendation System

The crop recommendation system in the *Fasal Mitra* project was developed to identify the most suitable crops based on a farmer's environmental conditions. These conditions included **soil nutrient levels (N, P, K)**, **pH**, **temperature**, **humidity**, and **rainfall**. The system leveraged machine learning to learn from historical data and suggest optimal crop choices, improving yield outcomes and supporting sustainable farming practices.

Given the nature of the problem—predicting a categorical output (crop name) from continuous environmental input variables—a **classification-based machine learning approach** was adopted. Multiple models were developed and evaluated to ensure both accuracy and generalizability across different scenarios.

Naive Bayes emerged as the most effective model for this task. It is a probabilistic classifier based on Bayes' Theorem with an assumption of independence among features. Despite its simplicity, it delivered remarkable accuracy—**99.55%**—making it an ideal candidate for fast, scalable predictions. Naive Bayes performed well due to the clean and structured nature of the dataset, where conditional probabilities aligned closely with actual crop distributions.

Random Forest Classifier was also employed as a robust ensemble method. It constructs multiple decision trees using different subsets of the training data and aggregates their predictions for final output. This approach reduces overfitting and captures complex feature interactions. With its ability to rank feature importance and handle non-linear relationships, Random Forest was among the top-performing models in terms of both accuracy and model interpretability.

Support Vector Machines (SVM) were implemented to explore the effectiveness of separating crop classes in high-dimensional feature space. SVM works by finding the optimal hyperplane that best divides the classes with maximum margin. It was particularly useful in handling non-linear boundaries in the dataset, although it required more fine-tuning and preprocessing (such as feature scaling) compared to other models.

K-Nearest Neighbors (KNN) was another intuitive model used. It predicts the crop by finding the 'k' most similar data points in the training set and assigning the most common crop among them. KNN performed well when appropriate values of 'k' were chosen, especially due to the clustered nature of agricultural data where similar conditions lead to similar outcomes.

Decision Tree Classifier was used for its interpretability and transparency. It splits the feature space using logical rules based on variables like pH, temperature, or NPK levels. The path from root to leaf represents a clear decision-making process that can be easily interpreted by farmers. While its standalone accuracy was lower than ensemble methods, it offered valuable insight into the structure of the data and the factors most influential in crop selection.

To evaluate the performance of all models, the project used metrics such as **Accuracy**, **Precision**, **Recall**, and **F1-score**. These metrics were derived from the **confusion matrix**, which helped in analyzing how well the models classified each crop category and where misclassifications occurred. High precision and recall ensured that the recommendations were not only correct but also reliable under varying input conditions.

In conclusion, the crop recommendation system utilized a well-rounded mix of classification algorithms to provide fast, interpretable, and accurate suggestions for optimal crop choices. The system's high performance—particularly from Naive Bayes and Random Forest—demonstrated its readiness for real-world application, enabling data-driven agricultural planning that can adapt to changing environmental inputs.

3.3 Model Training and Hyperparameter Tuning

To ensure high predictive accuracy and robustness, all machine learning models used in the *Fasal Mitra* project—both for crop recommendation and crop yield prediction—underwent a comprehensive training and hyperparameter tuning process. The goal was to optimize each model's performance while minimizing overfitting and maximizing generalization across unseen data.

Dataset Splitting

Both the **crop recommendation** and **crop yield prediction** datasets were divided into training and testing sets using an **80:20 split**. This allocation allowed the models to learn from the majority of the data while reserving a portion for unbiased performance evaluation.

Hyperparameter Tuning Process

Each machine learning algorithm was tuned by experimenting with multiple hyperparameter configurations. This tuning was essential to adjust the model's learning behavior and improve its predictive capabilities. Key hyperparameters that were optimized include:

- **Random Forest Classifier & Regressor**
  - Number of estimators (trees)
  - Maximum depth of trees
  - Minimum samples split and leaf size
- **Naive Bayes**
  - Assumed Gaussian distribution, no hyperparameter tuning required, but performance was compared across feature scaling methods.
- **Support Vector Machine (SVM)**
  - Kernel type (linear, RBF)
  - Regularization parameter (C)
  - Gamma value for non-linear kernels
- **K-Nearest Neighbors (KNN)**
  - Number of neighbors (k)
  - Distance metrics (Euclidean, Manhattan)
- **Gradient Boosting & Decision Tree**
  - Tree depth
  - Learning rate (for GBM)
  - Subsample and number of boosting rounds (for GBM)

Cross-Validation for Model Reliability

To avoid overfitting and validate the consistency of model performance across different data segments, **k-fold cross-validation** was employed (with $k = 5$ or 10 depending on the dataset size). This technique involved splitting the training data into k subsets and rotating the validation set, ensuring that each data point was tested once and trained on $k-1$ times.

Model Evaluation and Selection

After tuning, the models were evaluated on the test data using appropriate metrics:

- **For Classification (Crop Recommendation):** Accuracy, Precision, Recall, and F1-Score
- **For Regression (Yield Prediction):** Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score

Final model selection was based on the **best overall performance and stability** across these metrics. For instance:

- **Naive Bayes** achieved the highest classification accuracy (99.55%) for crop recommendation.
- **K-Nearest Neighbors Regressor** demonstrated superior performance in yield prediction with a low MAE and a high R^2 value of **0.9846**.

3.4 Model Evaluation and Performance

The effectiveness of machine learning models in the *Fasal Mitra* project was assessed using a comprehensive set of evaluation metrics tailored to the nature of each task—classification for crop recommendation and regression for crop yield prediction. This evaluation was crucial to ensure the reliability, accuracy, and practical applicability of the models in real-world agricultural decision-making.

Evaluation of Crop Recommendation Models

The crop recommendation system involved a **multi-class classification task**, where the goal was to correctly predict the most suitable crop based on environmental and soil features. The following performance metrics were used:

- **Accuracy** – Proportion of correct predictions among total predictions
- **Precision** – Ability of the model to return only relevant crops

- **Recall** – Ability to identify all relevant crops correctly
- **F1-score** – Harmonic mean of precision and recall, ensuring balance

After testing 10 different classification algorithms, the results were:

- **Naive Bayes** delivered the best overall performance with an accuracy of **99.55%**
- **Random Forest** achieved **99.32%**, making it a robust backup with slightly lower variance
- **SVM, KNN, and Decision Tree** also produced high accuracy (>96%) but were slightly less consistent across recall and F1-score

These results indicate that ensemble methods (Random Forest, Bagging) and probabilistic models (Naive Bayes) are highly effective for crop classification problems.

Evaluation of Crop Yield Prediction Models

For crop yield prediction, a **regression-based approach** was used. The following metrics were applied:

- **Mean Absolute Error (MAE)** – Average absolute difference between predicted and actual yields
- **Mean Squared Error (MSE)** – Average of squared errors, penalizing larger deviations
- **Root Mean Squared Error (RMSE)** – Square root of MSE, representing actual units of error
- **R^2 Score** – Proportion of variance in yield explained by the input features

Performance highlights include:

- **K-Nearest Neighbors Regressor (KNN)** showed the **lowest MAE of 4679.87** and the **highest R^2 score of 0.9846**, making it the most accurate model overall

- **Random Forest Regressor** and **Decision Tree Regressor** also demonstrated high accuracy with R^2 scores of **0.9806** and **0.9734**, respectively
- **Linear models** such as Ridge and Lasso performed significantly worse, suggesting the underlying agricultural data is non-linear and better captured by tree-based models

Interpretability and Practical Relevance

While models like Random Forest and Gradient Boosting delivered strong performance, **Decision Trees** were especially valuable for interpretation. Their clear decision paths made it easier to understand which factors—such as pH, rainfall, and nutrient levels—had the most impact on predictions.

Conclusion

In conclusion, the machine learning models deployed in *Fasal Mitra* demonstrated high accuracy and reliability for both crop recommendation and yield prediction tasks. These models were:

- **Trained and tuned rigorously**
- **Evaluated across multiple metrics**
- **Tested for real-world viability through structured testing and comparison**

Their successful integration into a web-based platform lays the foundation for a scalable smart farming solution that empowers farmers to make better decisions, optimize inputs, and improve agricultural productivity.

The next chapter outlines the **development of this integrated web application**, which brings these predictive capabilities directly to the end user.

CHAPTER: IV
Application Development

4.1 Frontend Development

The frontend of the application was developed with the primary goal of creating a simple, responsive, and visually appealing interface. It was built using **HTML, CSS, and JavaScript**, which together provided the necessary structure, styling, and interactivity for the application to function efficiently.

HTML (HyperText Markup Language) was used to define the structural layout of the web pages. It facilitated the creation of essential components such as input forms, interactive buttons, and tables or charts for displaying the output. Serving as the backbone of the frontend, HTML enabled an organized and intuitive layout where farmers could easily navigate through the application and input critical information like region, soil pH, and rainfall levels. These inputs would then trigger real-time predictions and crop recommendations using the integrated machine learning models.

CSS (Cascading Style Sheets) was employed to enhance the visual appeal and layout consistency of the application. The use of responsive design techniques ensured that the interface automatically adapted to different screen sizes, making it accessible on both desktops and mobile devices. Considering that many users may access the platform via smartphones, the design emphasized a mobile-friendly experience—characterized by clean visuals, properly scaled fonts, and strategically placed buttons. Special attention was given to color schemes and spacing to maintain readability and ensure a user-friendly interface.

JavaScript was used to bring interactivity and dynamic behavior to the frontend. It enabled real-time updates based on user inputs. For instance, when a farmer entered specific data like soil temperature or rainfall, JavaScript dynamically processed the input and triggered the display of predictions or recommendations. Additionally, JavaScript allowed for the integration of interactive elements such as graphs and charts to visually represent key agricultural data. These visual tools made it easier for users to understand complex information and make informed decisions based on easily interpretable outputs.

4.2 Backend Development

The backend of the *Fasal Mitra* application was developed using **Python**, chosen for its simplicity, scalability, and rich ecosystem of libraries that support machine learning, data processing, and web development. Python provided the necessary flexibility to seamlessly integrate trained ML models and implement the business logic that powers the application.

To build the server-side functionality, the **Flask framework** was used. Flask is a lightweight Python web framework that enabled the rapid development of RESTful APIs. These APIs act as the communication bridge between the frontend interface and the machine learning models. When the user submits input—such as soil conditions, temperature, or rainfall—via the web interface, Flask routes the data to the relevant ML model, which processes the input and returns a prediction or crop recommendation to be displayed on the frontend.

One of the critical tasks in backend development was **machine learning model integration**. The models for crop recommendation and yield prediction were developed and trained using Python libraries like **Scikit-learn** on historical agricultural datasets. These models were then serialized using **joblib** and loaded into the Flask application to support real-time inference. Upon receiving user data, the backend invokes the appropriate model, processes the input, and returns an accurate output almost instantly.

In addition to model execution, the backend also handled **data preprocessing** tasks to ensure the inputs were clean and suitable for prediction. Libraries such as **Pandas** were used for validating input data, handling missing values, and performing necessary feature transformations. This preprocessing step was essential to maintain the integrity of the predictions and ensure the models performed reliably.

While the current deployment is local or standalone in nature, the system architecture has been designed to be **scalable and deployment-ready**, allowing for easy transition to cloud platforms in the future, if required.

4.3 Integration of Machine Learning Models

The integration of machine learning models into the *Fasal Mitra* web application forms the core of its intelligent functionality. This process allows users to interact with trained models in real-time through a simple and responsive web interface.

Both the **crop recommendation** and **yield prediction** models were developed using Python's **Scikit-learn** library and saved using **Joblib** for efficient reuse. These serialized models were embedded into a backend system built with **Flask**, which handles user input, processes it, and returns prediction results.

When a user submits data via the web interface—such as soil nutrient levels, temperature, humidity, or rainfall—the input undergoes the following steps:

1. **Validation and preprocessing** using Python libraries like **Pandas** and **NumPy**.
2. Data is passed into the appropriate machine learning model (classification for crop recommendation or regression for yield prediction).
3. The model processes the input and generates predictions.
4. The output is then sent back to the frontend for display.

The application is **deployed and hosted on Render**, a free cloud platform, enabling online accessibility without the need for local hosting. Render handles the backend routing and ensures that predictions can be generated on-demand, even when accessed from mobile devices or remote locations.

This integration ensures a seamless connection between user input and machine learning logic, providing farmers with fast, reliable, and personalized insights that are accessible from any internet-enabled device.

4.4 Features of the Application

The *Fasal Mitra* web application was developed with the objective of assisting farmers in making accurate, data-driven decisions related to crop planning. The platform integrates user-friendly web interfaces with machine learning models to deliver actionable insights. The key features implemented in the application include:

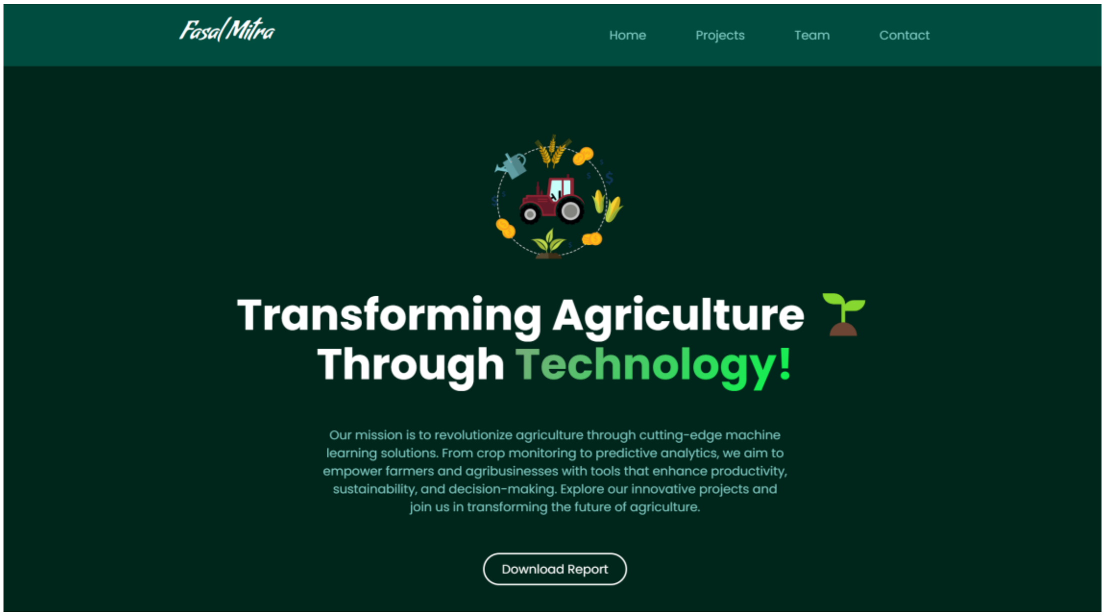

Figure 4.1 *Homepage screenshot of Fasal Mitra*

Crop Yield Prediction

The application enables users to input essential agricultural data such as soil nutrient levels, temperature, humidity, rainfall, and crop type. Based on this input, the integrated regression-based machine learning models predict the expected crop yield. This helps farmers estimate potential output in advance, allowing for better planning and efficient resource allocation. The interface is designed to be intuitive, so users can easily enter their data and receive quick, accurate predictions.

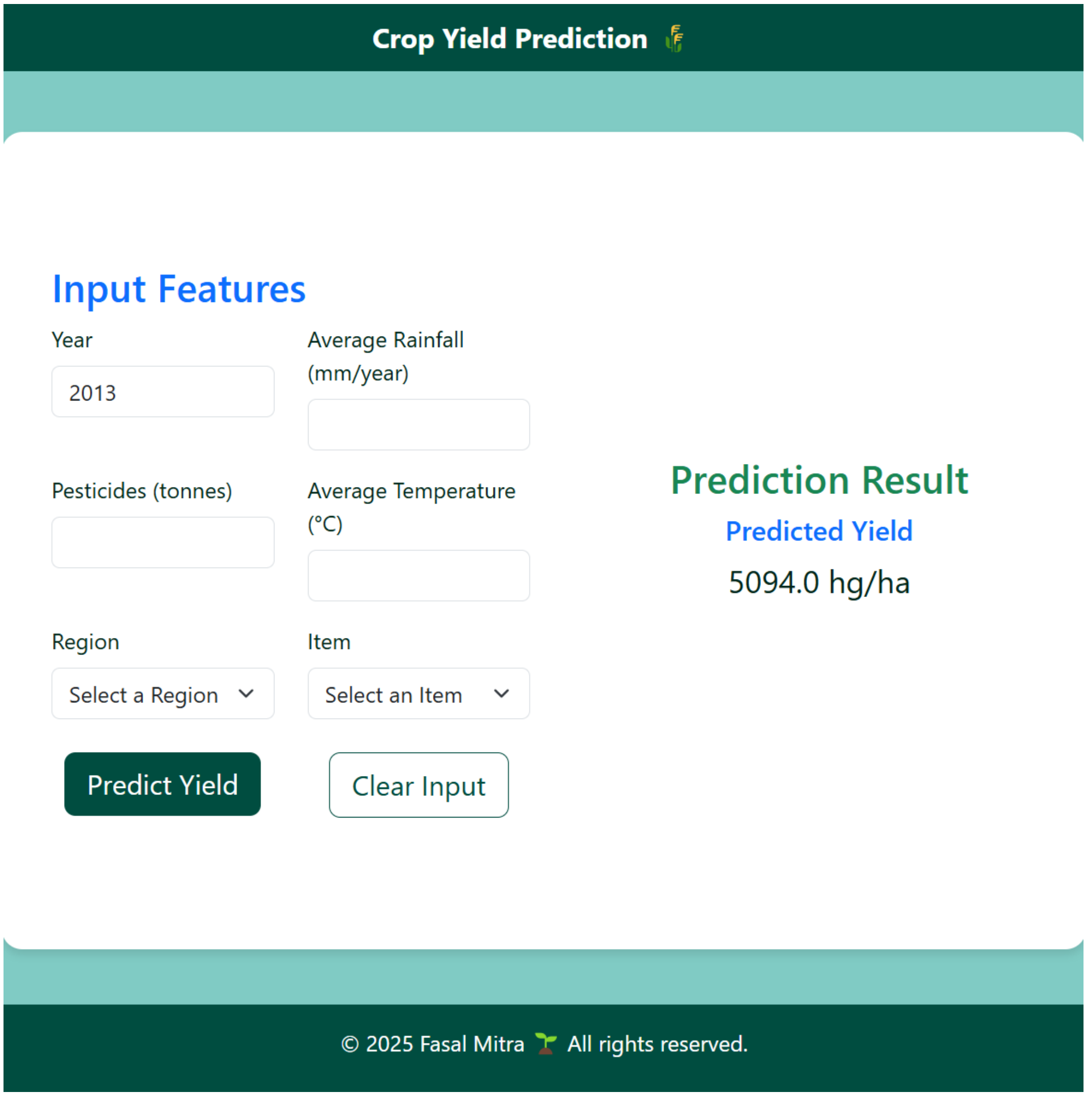

Figure 4.2 *Crop Yield Prediction UI*

Crop Recommendation:

Using classification algorithms, the application analyzes user-provided data such as nitrogen, phosphorus, potassium (NPK) values, pH level, temperature, humidity, and rainfall. Based on these features, it suggests the most suitable crop for cultivation under the given environmental conditions. The results are displayed in a straightforward format, allowing farmers to confidently select crops that are best suited to their region and soil profile.

The entire system is accessible via a web browser and is optimized for both desktop and mobile views. The clean, responsive design ensures that even users with minimal technical background can navigate the platform and benefit from its features.

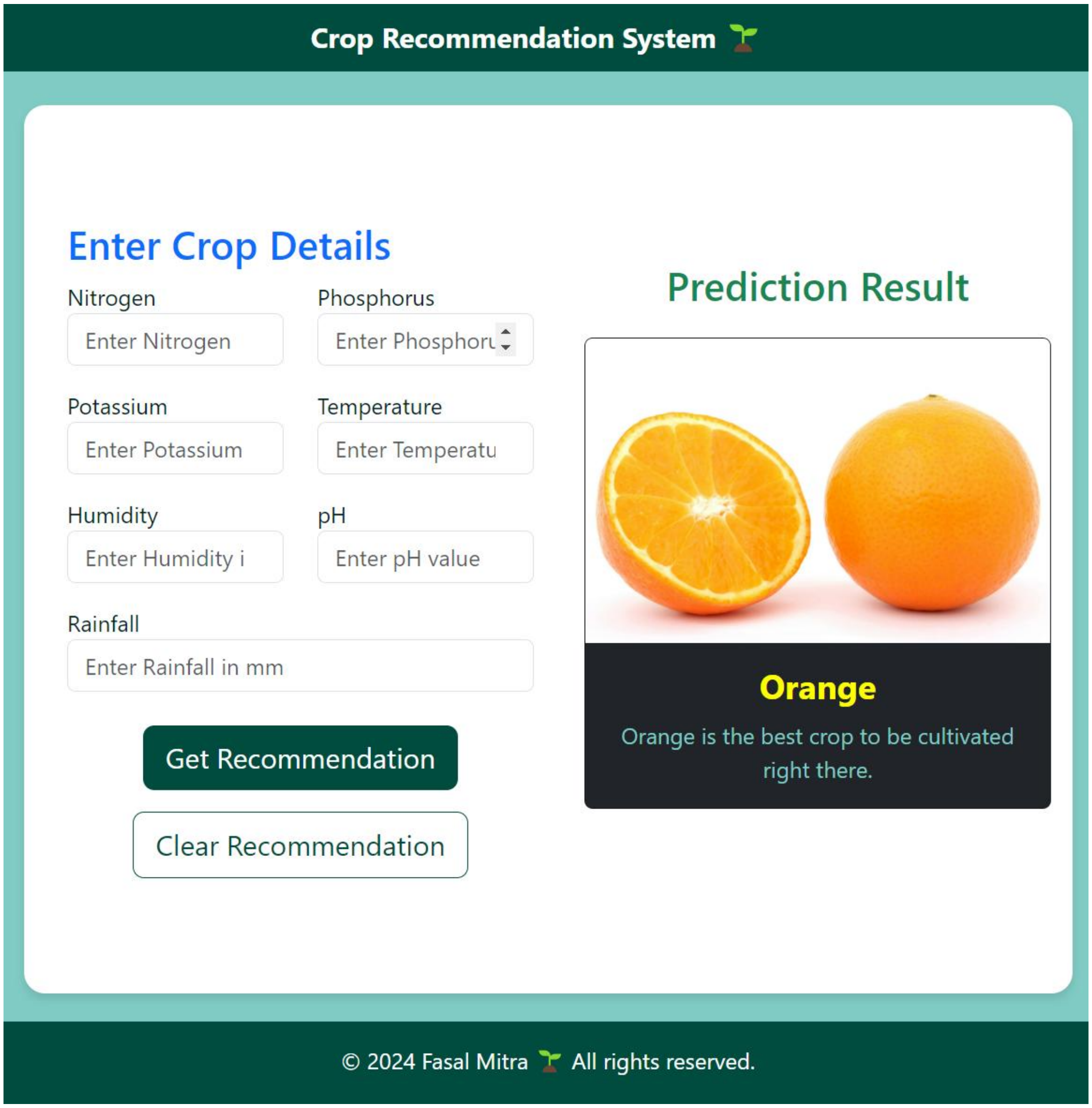

Figure 4.3 *Crop Recommendation UI*

CHAPTER: V
Results & Discussion

5.1 Model Accuracy

To evaluate the effectiveness of the machine learning models integrated into the *Fasal Mitra* application, several classification and regression algorithms were tested. The performance of each model was assessed using standard metrics to determine their accuracy and reliability in solving the respective prediction tasks.

a) Classification Models Performance (Crop Recommendation System)

For crop recommendation, the task was treated as a **multi-class classification problem**, where models predicted the most suitable crop based on soil and environmental conditions. The performance of each model was evaluated using four metrics: **Accuracy, Precision, Recall, and F1-score**.

The table below presents a comparative analysis of the classification models used:

TABLE 5.1 *CLASSIFICATION MODELS PERFORMANCE*

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.9955	0.9958	0.9955	0.9954
Random Forest	0.9932	0.9937	0.9932	0.9932
Bagging	0.9864	0.9867	0.9864	0.9864
Decision Tree	0.9841	0.9845	0.9841	0.9841
Gradient Boosting	0.9818	0.9843	0.9818	0.9819
Support Vector Machine (SVM)	0.9682	0.9715	0.9682	0.9680
Logistic Regression	0.9636	0.9644	0.9636	0.9635

Model	Accuracy	Precision	Recall	F1-score
K-Nearest Neighbors (KNN)	0.9591	0.9654	0.9591	0.9590
Extra Trees	0.9000	0.9044	0.9000	0.9001
AdaBoost	0.0955	0.0514	0.0955	0.0575

From the results, **Naive Bayes** emerged as the best-performing classification model, achieving the highest accuracy of **99.55%**. It was followed closely by **Random Forest (99.32%)** and **Bagging (98.64%)**. These results demonstrate that ensemble and probabilistic models are particularly effective for this multi-class agricultural classification task. On the other hand, **AdaBoost** significantly underperformed, making it unsuitable for this use case.

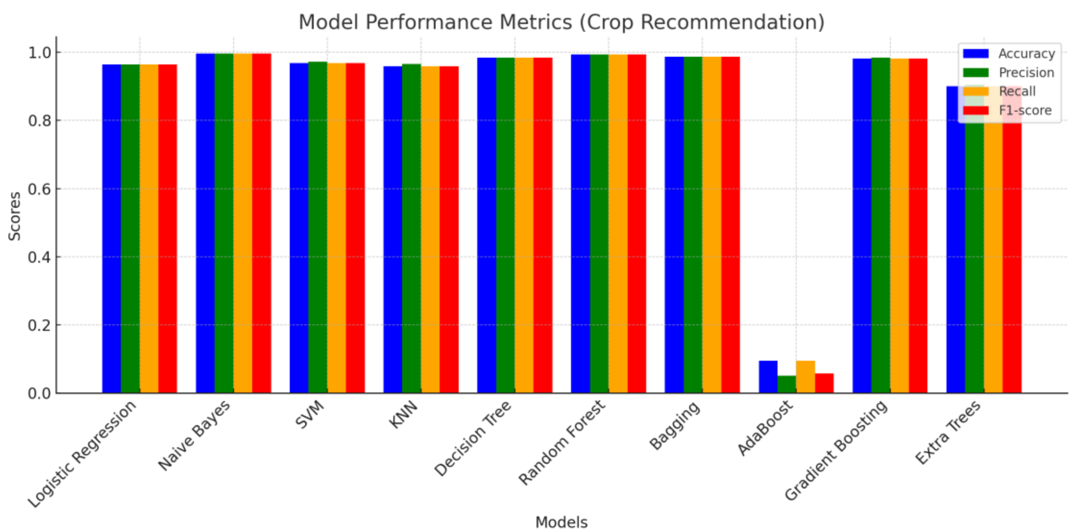

Figure 5.1 *Classification Model Performance*

b) Regression Models Performance (Crop Yield Prediction)

For crop yield prediction, the problem was approached using **regression models** that estimate numerical yield values based on user inputs. The models were evaluated using **MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error)**, and **R^2 Score**.

TABLE 5.2 *REGRESSION MODELS PERFORMANCE*

Model	**MAE**	**MSE**	**RMSE**	**R^2 Score**
Lasso	29883.8307	1.82e+09	42631.8842	0.7473
Ridge	29852.8541	1.82e+09	42634.5089	0.7473
Decision Tree	5265.9318	1.91e+08	13823.4912	0.9734
K-Nearest Neighbors (KNN)	4679.8747	1.11e+08	10516.2354	0.9846
Random Forest	5155.3739	1.40e+08	11827.2896	0.9806
Gradient Boosting	25520.8272	1.76e+09	41914.7970	0.7557

Among the regression models, **KNN Regressor** outperformed all others, achieving the **lowest MAE (4679.87)** and the **highest R^2 score (0.9846)**, making it the most suitable model for accurate yield prediction. **Random Forest** and **Decision Tree** regressors also performed well, while linear models like **Lasso** and **Ridge** showed poor performance, indicating they were not suitable for the non-linear agricultural data used in this project.

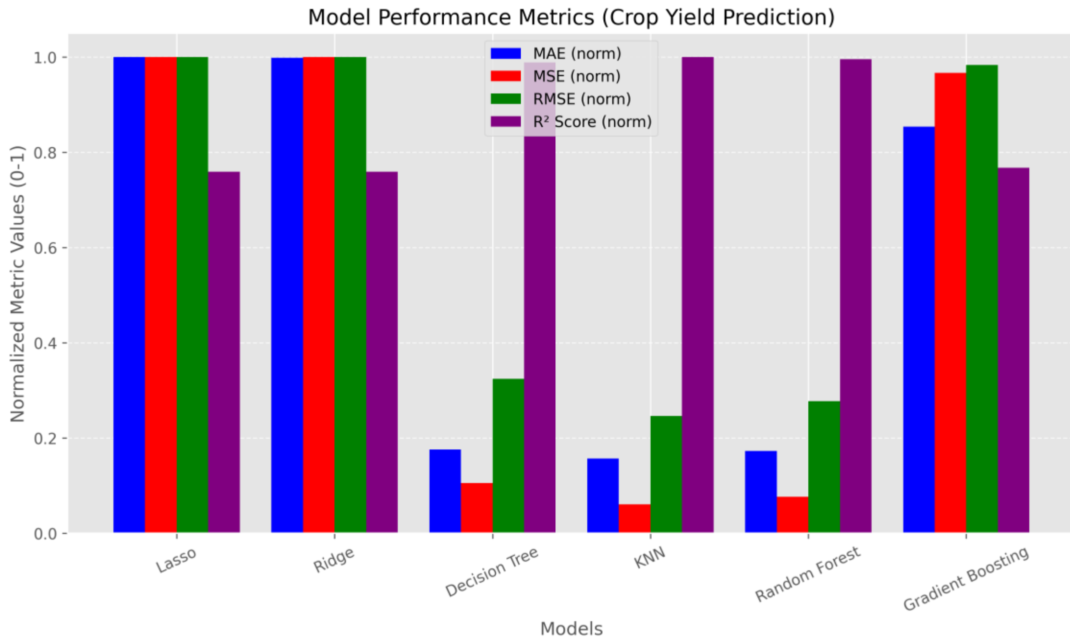

Figure 5.2 *Regression Model Performance*

5.2 User Testing

To evaluate the usability and practical effectiveness of the *Fasal Mitra* web application, user testing was conducted involving both farmers and agricultural experts. The goal was to assess how well the system aligned with real-world agricultural practices, the clarity of the user interface, and the accuracy of its outputs.

Farmer Feedback

Farmers, being the primary end-users, were guided through key functionalities including crop recommendation, yield prediction, and data entry. Most participants reported that the **interface was simple, intuitive, and easy to navigate**. They particularly appreciated the **clarity of the outputs** presented through charts and result tables. The **mobile accessibility** of the platform was highlighted as a significant advantage, as many users accessed the system through smartphones. A common suggestion was to include **regional language support** to improve accessibility in rural areas where English fluency is limited.

Agricultural Expert Feedback

Agricultural scientists and field officers reviewed the application from a technical perspective. They validated the **relevance and accuracy** of the crop and yield predictions, observing that the outputs were consistent with historical agricultural data trends. However, they noted the importance of accounting for **regional environmental variations**, and recommended integrating **more localized datasets** for improved precision.

Common Observations

- **Positive Feedback:**
  The visualizations and simplified data presentation were praised for making complex insights accessible, even for users with minimal technical expertise.
- **Areas for Improvement:**
  Farmers expressed interest in additional features such as **region-specific pest control advice**, tracking crop growth stages through regular data inputs, and support for **less common crops**.

Overall, the user testing process affirmed that *Fasal Mitra* is a **functional and user-friendly platform**, with a strong foundation for future enhancements based on user needs.

5.3 Discussion

The development, deployment, and evaluation of the *Fasal Mitra* application revealed several important insights. While the system demonstrated clear potential in assisting farmers with data-driven decisions, it also highlighted specific challenges and areas for improvement that are critical for its future evolution.

Data Bias

One key challenge was the **bias in available datasets**. The machine learning models performed well on commonly cultivated crops such as rice and wheat due to the abundance of training data. However, the predictions for **region-specific or niche crops** were less reliable. Addressing this imbalance will require **collecting more diverse and inclusive datasets**, particularly from underrepresented regions and crop types.

Limited Datasets

Despite efforts to compile robust datasets, there were limitations in terms of **data availability and granularity**. Real-time variables such as **soil moisture, pest outbreaks, or hyper-local weather conditions** were not included due to lack of accessible data. Incorporating **IoT devices** or collaborating with **agricultural organizations** could help enrich the system with real-time, dynamic inputs in the future.

Scalability Challenges

The current backend setup, including the use of **SQLite**, is sufficient for initial development and small-scale testing. However, as user adoption grows, the system may encounter **performance and concurrency issues**. Migrating to **cloud-based databases and platforms** such as AWS or Google Cloud will be essential to ensure scalability and long-term reliability.

Need for Customization

User feedback suggested a demand for **greater personalization and advisory features**, such as:

- Pest and disease alerts
- Seasonal crop rotation reminders

- Localized best practices

  These enhancements would require integrating additional models and external data sources tailored to regional agricultural practices.

Local Adaptability

A critical factor observed during testing was the **regional adaptability of the recommendations**. In areas with **extreme weather patterns** or unique soil conditions, the models' predictions were sometimes less effective. Incorporating **hyper-local weather data, detailed soil profiling**, and **satellite imagery** could significantly improve the relevance and reliability of the outputs.

In conclusion, *Fasal Mitra* successfully demonstrated the potential of machine learning in supporting smart farming. The insights gained through user testing and system performance evaluation provide a **clear roadmap for future upgrades**. Addressing data diversity, regional adaptability, and platform scalability will be key to transforming the application into a powerful, inclusive, and widely usable tool for modern agriculture.

CHAPTER: VI
Conclusion & Future Scope

6.1 Conclusion

The development and implementation of the *Fasal Mitra* web application mark a meaningful step toward the modernization of agriculture through the integration of **machine learning and data-driven decision-making**. This project demonstrated how technology can effectively address major farming challenges, particularly in areas such as **crop yield prediction** and **crop recommendation**.

By utilizing robust agricultural datasets and applying supervised machine learning models, the application successfully provided farmers with accurate, timely, and personalized insights. These included predicted crop yields and suitable crop suggestions based on environmental and soil parameters, helping users optimize their farming strategies.

The frontend of the application was developed using **HTML, CSS, and JavaScript**, designed to offer a clean and responsive user experience. The backend, powered by **Python and Flask**, handled model integration and input processing with high efficiency. The overall system was built with simplicity in mind, ensuring accessibility for farmers—even those with limited technical expertise.

Feedback collected from **farmers and agricultural experts** during user testing confirmed the application's usability and effectiveness. Users appreciated the intuitive interface, informative visualizations, and the clarity of the output, all of which made it easier to interpret complex data. The positive reception reaffirmed the value of incorporating predictive analytics into everyday agricultural decision-making.

However, the project also highlighted key limitations, such as **data bias**, limited **dataset diversity**, and the absence of real-time environmental data. These issues occasionally affected the accuracy of predictions, particularly for less common crops or underrepresented regions. Despite these challenges, the experience provided crucial insights and a foundation for future enhancements.

In conclusion, *Fasal Mitra* has shown strong potential to enhance the efficiency, profitability, and sustainability of farming practices. It empowers farmers with reliable insights, promotes smarter resource utilization, and sets the stage for more **scalable, adaptable, and intelligent agricultural solutions** in the future.

6.2 Future Scope

While the *Fasal Mitra* application has successfully addressed several critical challenges in modern agriculture through machine learning-driven crop recommendation and yield prediction, there remains substantial scope for enhancement. Future development efforts aim to expand the platform's capabilities, increase precision, and improve usability for a broader range of users and environments.

Integration with IoT Devices

A key direction for future work is the integration of **IoT (Internet of Things)** devices such as **soil moisture sensors, temperature probes, and weather monitoring stations**. These devices would enable real-time data collection, providing continuous updates on environmental and soil conditions. With this dynamic input, the machine learning models can generate more context-aware predictions and recommendations. Potential applications include:

- Real-time alerts for **irrigation scheduling**
- Warnings about **adverse weather conditions**
- Notifications for **fertilizer application timing**

Such features would empower farmers with highly responsive, data-driven decision support, enhancing productivity and resource efficiency.

Inclusion of Pest and Disease Management

During user testing, a commonly requested feature was the inclusion of **pest and disease control recommendations**. In future iterations, the platform can be enhanced with:

- **Image recognition models** to detect signs of pest infestation or disease from crop images
- **Predictive analytics** based on regional pest behaviour and seasonal patterns

This functionality would enable timely interventions, reduce pesticide overuse, and minimize crop losses—resulting in improved crop health and quality.

Enhancing Regional Adaptability

To improve the accuracy and relevance of recommendations across diverse geographic locations, future work will involve expanding the dataset to include **region-specific and hyper-local information**. Collaborations with local agricultural agencies and research institutions could help gather detailed regional data such as:

- Local soil characteristics
- Microclimatic weather forecasts
- Crop performance records by region

This would make the system more adaptable and effective in diverse agro-climatic zones.

Cloud-Based Infrastructure for Scalability

As the user base grows, migrating the backend infrastructure to **cloud platforms** like **AWS, Microsoft Azure, or Google Cloud** will be essential. Cloud deployment will allow the system to:

- Handle larger datasets and concurrent user traffic
- Store and process real-time data
- Scale seamlessly without performance bottlenecks

This upgrade will support the continued expansion of *Fasal Mitra* and ensure long-term reliability.

Multi-Language Support

To improve accessibility for farmers in rural and linguistically diverse regions, the application will be expanded to support **multiple regional languages**. This will help remove language barriers, encourage adoption among non-English-speaking users, and make the application more inclusive and farmer-friendly.

These proposed enhancements form a roadmap for evolving *Fasal Mitra* into a **fully intelligent, responsive, and regionally adaptable smart farming solution**. With ongoing improvements, the application can become a comprehensive agricultural assistant that contributes to greater efficiency, sustainability, and prosperity in the farming community.

References:

[1] M. Kamilaris and F. X. Prenafeta-Boldú, "Machine learning applications in agriculture: A review," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[2] M. Osman, N. Abdalla, and N. A. Elagib, "Climate variability and change affect crops yield under rainfed conditions: A case study in Gedaref state, Sudan," *Agronomy*, vol. 11, no. 9, p. 1680, 2021.

[3] S. P. Raja et al., "Crop prediction based on characteristics of the agricultural environment using various feature selection techniques and classifiers," *International Journal of Research Publication and Reviews*, vol. 3, no. 1, pp. 102–112, 2022.

[4] S. Ali, M. A. Qureshi, and N. Javaid, "Smart farming using machine learning

and deep learning techniques," *Computers and Electronics in Agriculture*, vol. 200, p. 107109, 2022.

[5] D. Singh and A. S. Reddy, "Crop prediction using machine learning: An ensemble approach," *Agronomy*, Special Issue, MDPI, 2021.

[6] M. Altalak, M. Ammad-Uddin, and A. Alzahrani, "Smart agriculture applications using deep learning technologies: A survey," *Applied Sciences*, vol. 12, no. 12, p. 5919, 2022.

[7] A. R. Dhekekar and S. V. Dandavate, "Dimensionality reduction for agriculture data using PCA," *Journal of Critical Reviews*, vol. 7, no. 6, pp. 861–865, 2020.

[8] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of precision agriculture," *Computers and Electronics in Agriculture*, vol. 157, pp. 218–231, 2019.

[9] Y. Shen et al., "Optimizing machine learning for agricultural data using Bayesian optimization," *Agronomy*, vol. 12, no. 2, p. 456, 2022.

[10] S. Ghosal et al., "Explainable AI for agriculture: Use of SHAP to interpret model predictions," *Computers and Electronics in Agriculture*, vol. 178, p. 105780, 2020.

[11] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2011.

[12] Atharva Ingle, "Crop Recommendation Dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

[13] Rishi Patel, "Crop Yield Prediction Dataset," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/patelris/crop-yield-prediction-dataset