AI-Driven Agronomy: Enhancing Farm Productivity using AI

Project report submitted to the Amrita Vishwa Vidyapeetham in partial fulfilment of the requirement for the Degree of

BACHELOR of TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)

Submitted by

Thota Bhuvana Chandra AM.EN.U4AIE20170
Kona Mourya Sai Chandra AM.EN.U4AIE20140
N Abhinay reddy AM.EN.U4AIE20149

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM
(Estd. U/S 3 of the UGC Act 1956)
AMRITAPURI CAMPUS

KOLLAM -690525

MAY 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)

AMRITA VISHWA VIDYAPEETHAM
(Estd. U/S 3 of the UGC Act 1956)

Amritapuri Campus

Kollam -690525

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"AI-Driven Agronomy: Enhancing Farm Productivity using AI"** submitted by **Thota Bhuvana Chandra (AM.EN.U4AIE20170), Kona Mourya Sai Chandra (AM.EN.U4AIE20140) and N Abhinay Reddy (AM.EN.U4AIE20149)**, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence) from Amrita Vishwa Vidyapeetham, is a bonafide record of the work carried out by them under my guidance and supervision at Amrita School of Computing, Amritapuri during Semester 8 of the academic year 2023-2024.

Project Guide: Dr.Aiswarya S Kumar

Project Coordinator: Ms Asha Ashok

Place : Amritapuri
Date : 30 May 2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE)

AMRITA VISHWA VIDYAPEETHAM
(Estd. U/S 3 of the UGC Act 1956)

Amritapuri Campus

Kollam -690525

श्रद्धावान् लभते ज्ञानम्

DECLARATION

We, **Thota Bhuvana Chandra (AM.EN.U4AIE20170), Kona Mourya Sai Chandra (AM.EN.U4AIE20140)) and N Abhinay Reddy (AM.EN.U4AIE20149)** hereby declare that this project entitled **"AI-Driven Agronomy: Enhancing Farm Productivity using AI"** is a record of the original work done by us under the guidance of **Dr.Aiswarya S Kumar**, Department of Computer Science and Engineering , Amrita Vishwa Vidyapeetham, that this work has not formed the basis for any degree/diploma/associationship/fellowship or similar awards to any candidate in any university to the best of our knowledge.

Place : Amritapuri

Date : 30 May 2024

Signature of the student | Signature of the Project Guide

Acknowledgements

First and foremost, we thank our Chancellor, Satguru Sri Mata Amritanandamayi Devi (AMMA), whose vision and guidance have given us the opportunity to pursue this enriching project.

We are extremely grateful to Dr.Aiswarya S Kumar for her guidance. Working under her tutelage has been a privilege, and her insights and expertise have greatly enriched our project experience.

We feel extremely grateful to our reviewer, Dr.Divya Udayan J for her feedback and encouragement which helped us to complete the project.

We would like to thank our project coordinator, Ms Asha Ashok for her invaluable guidance and support throughout this project.

We would also like to thank the entire fraternity of the Department of Computer Science and Engineering (Artificial Intelligence), Amrita School Of Computing, and extend our heartfelt gratitude to all the faculty members, research staff, faculty associates, and office personnel. Their unending support and encouragement have made this project possible. We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project. Finally, we are grateful for the opportunity to work on this project, as well as the guidance and support we have received throughout the process. We hope that our work will help advance research and development efforts in our field.

Abstract

Agriculture is crucial to the economy, contributing considerably to GDP and employing a large workforce. Nonetheless, precisely projecting food production and demand remains a difficult task. Traditional prediction approaches are time-consuming and prone to errors, resulting in inefficient resource allocation and financial losses. To solve these issues, we offer an innovative platform that uses machine learning algorithms to predict the best crops for production based on environmental conditions, as well as forecast market demand and pricing patterns for those crops.It contain four main modules these are The Crop Prediction module uses machine learning and historical data to forecast crop growth, helping farmers make informed decisions to increase yields. The Crop Price Prediction module applies predictive analytics to predict market trends and price changes, assisting farmers in strategic planning. The Fertiliser Prediction module use data-driven algorithms to predict the best fertiliser mixtures and application rates depending on soil nutrients, crop needs, and environmental factors. The Crop Yield Prediction module uses machine learning and historical data to forecast yields prior to harvest, assisting farmers in optimising resource allocation and production plans.A tiny gadget with integrated sensors evaluates critical soil properties and delivers real-time data on environmental conditions and soil health. This real-time data enables farmers to precisely monitor crop growth and perform appropriate interventions, such as irrigation scheduling and soil treatment adjustments, to increase crop quality and production. This technology enables farmers to make data-driven decisions and promotes sustainable agricultural practices in a tough and changing environment by integrating real-time monitoring with predictive capabilities.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Agriculture has long been recognised as a significant source of human survival, serving as a major vocation and industrial sector in India. Farmers have always depended on visual observation and pesticide-free procedures to diversify their crops and livestock. Today's shifting weather and resource constraints put food security at risk, with the agricultural sector's GDP contribution expected to fall from 3.4% in 2022 to 2.8% in 2023.This decline directly affects 80 percent of rural farmers depending on crop income. To address this, India can enhance crop profitability and quality through technological advancements like machine learning. This enables crop forecasting, crop price estimation, fertilizer prediction and crop yield estimation based on the atmospheric, soil parameters, and previous price trends specific to the farmland. In our globalized world with unpredictable climates, machine learning employs advanced algorithms to analyze extensive agricultural data. By processing historical and real-time information on weather, soil, harvest patterns, and more, machine learning models make predictions and offer valuable recommendations. This allows farmers to optimise crop selection, plan irrigation, manage pests, and effectively allocate resources, leading in enhanced productivity and profitability.

Machine learning-based algorithms for crop forecasting, price estimate, fertiliser prediction, and yield estimation provide considerable advantages by discovering patterns and correlations in large datasets that humans may overlook. These models identify intricate correlations between variables by training algorithms on years of historical agricultural data and incorporating real-time inputs. As a result, they provide exact predictions and practical guidance, allowing farmers to adjust to environmental changes and make educated decisions.

Machine learning may alter agricultural operations by predicting crops, estimating prices, recommending fertilisers, and projecting yields. Accurate predictions and actionable insights help farmers reduce losses, enhance yields, and contribute to global food security. This paper investigates several machine learning algorithms for crop prediction and price estimate, as well as the datasets on which they are based and their practical applications. Furthermore, the produced code is available via a website, allowing farmers and other stakeholders quick access to these resources. The platform also anticipates crop profitability by projecting market demand and pricing patterns, as well as recommending appropriate fertilisers, with the goal of increasing profitability for all stakeholders in the agricultural business.

This system goes beyond basic prediction by providing real-time monitoring of important factors such as NPK levels, moisture content, humidity, and pH levels using sophisticated sensors incorporated into a user-friendly device. Farmers may reduce resource waste and establish the greatest circumstances for crop growth by refining soil management practices, irrigation scheduling, and fertiliser usage. Our Crop Monitoring System gives an excellent perspective of the fields, allowing farmers to make more informed decisions that improve profitability, sustainability, and production. Join us in revolutionising agriculture, where tradition meets technology to promote success and development.

Chapter 2

Problem Definition

2.1 Problem Definition

The agricultural sector confronts enormous challenges in properly forecasting crop yields, demand, and fertiliser needs. Conventional methods of estimating agricultural outputs and demand are time-consuming and susceptible to errors, leading to inefficient resource allocation and financial losses. Hence, there is a pressing need for advanced solutions that can forecast the optimal crop selection, predict yield quantities, determine the appropriate fertilizer usage based on varying environmental factors, and anticipate crop profitability. By leveraging cutting-edge technology and data-driven insights, we can revolutionize the way agricultural decisions are made, optimizing resource utilization and maximizing returns for farmers and stakeholders alike.

2.2 Objective

To create an advanced intelligent platform, we aim to harness the power of machine learning to recommend optimal crop and fertilizer choices tailored to specific environmental conditions also helps the farmer to know the amount for crop yield which helps farmer to plan his crop cycle accordingly. This innovative solution will not only consider the climate, soil quality, and other relevant factors but will also incorporate predictive analytics to anticipate market demand and pricing trends for the identified crops.

By seamlessly integrating environmental data with market insights, our platform aspires to empower farmers and agricultural stakeholders with strategic decision-making tools. Through real-time monitoring of key agronomic parameters, such as NPK levels, moisture content, humidity, and pH levels, our platform equips farmers with the tools they need to optimize crop health and productivity. Integrated sensors, seamlessly integrated into a user-friendly device, provide instant access to critical data, enabling farmers to make timely adjustments to fertilizer applications, irrigation schedules, and soil management practices. This level of precision agriculture ensures that resources are utilized efficiently, minimizing waste and environmental impact while maximizing yield potential.

Chapter 3

Related Work

Paper	Title/Year	Problem addressed	Contributions	Limitations	Open-Problems
[1] Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications Authors: D. Elavarasan and P.M. D. Vincent	IEEE Access, vol. 8,pp. 86886-86901, Year - 2020	An RNN-based feature processing combined with Deep Recurrent QNetwork model-based self-experimental analysis is constructed to forecast the crop yield. The forecast is done based on major climatic factors, and soil parameters datasets extracted from the Indian Meteorological Department's portal.	The results of DRQN model was compared with other ANN, and BAN models using certain evaluation metrics like error, and variance score and it outperformed all of them with 94 Percent accuracy. The probability density of actual and predicted yield was also measured.	The models are not able to create a direct non-linear or linear mapping between the raw data and crop yield values. The performance of the models highly relies on the quality of the extracted features. The model requires a large amount of training with a huge dataset.	Working on improving accuracy and creating a dataset for implementation

Paper	Title/Year	Problem addressed	Contributions	Limitations	Open-Problems
[2] Crop and Nutrient Recommendation System Using Machine Learning for Precision Agriculture Authors: Shabari Shedthi B1, Vidyasagar Shetiy2, Anusha1, Rakshitha R Shetiy1, Anisha Shetiy1, B.A Divyashree Alva	International Conference on Artificial Intelligence and Data Engineering (AIDE) Year - 2022	The paper evaluates the performance of the crop recommendation system using a 10-fold cross-validation.	The ensemble technique combines the predictions of three ml models: random forest, naive Bayes, and linear SVM. The ensemble technique used in the paper is called majority voting. With majority voting, the system recommends the crop that is most frequently recommended by the three machine learning models.	The system recommends a crop that is most frequently recommended by the three machine learning models. For example, if two models recommend crop A and one model recommends crop B, the system will recommend crop.	An analysis of multiple trials proved the model to have achieved a remarkably high mean precision of 96.3.

Paper	Title/Year	Problem addressed	Contributions	Limitations	Open-Problems
[3] Intelligent Crop Recommendation System using Machine Learning Authors: Priyadharshini A, Aayush Kumar, Swapneel Chakraborty	(ICCMC 2021) Year - 2021	According to the survey results, a significant 80	Developed a machine-learning model aimed at forecasting the yields of various crops. From a dataset containing historical weather data and soil information from diverse regions across India. The implementation of the neural network was facilitated using the Keras module. The model was designed as a sequential one, comprising 3 input layers and 15 output layers.	More time-consuming as the model's predictive capability for sustainability was assessed for each crop based on the three input parameters. Consequently, this assessment provided valuable insights into which crops were likely to yield better results.	The outcomes of the evaluation demonstrated the model's remarkable accuracy, with a prediction rate of 90 percent when it came to forecasting crop yields. This success in accurate prediction underscores the potential utility and effectiveness of the developed model for assisting farmers.

Paper	Title/Year	Problem addressed	Contributions	Limitations	Open-Problems
[4] Light GBM Algorithm based Crop Recommendation by Weather Detection and Acquired Soil Nutrients [2022] Authors: Jaichandran R1, T.Murali Krishna2, Sri Harsha Arigela3, Ramakrishnan	International Conference on Power, Energy, Control and Transmission Systems (ICPECTS) 2022	The complex algorithms underlying its system are designed to determine the optimal agricultural selection, contingent upon scrutinizing fluctuations in meteorological patterns as well as quantities of chemicals and minerals present within the substratum's composition.	The research also involved the evaluation of the system using real-world crop yield data from India. This empirical analysis served as strong evidence supporting the system's reliability and usefulness for farmers. Notwithstanding the complex issues, the researchers confronted in the duration of their inquiry.	The data they collected was noisy and incomplete, which presented difficulties in effectively training the machine learning algorithm. Additionally, the extensive number of features in the dataset posed another hurdle, making efficient training of the algorithm a demanding task.	The outcome garnered from the assessment proved notable, signifying that the mechanism procured a precision percentage of 95. This demonstrates the mechanism's potential to be an invaluable implement for cultivators endeavoring to elevate harvest selection and reform their bucolic routines.

Paper	Title/Year	Problem addressed	Contributions	Limitations	Open-Problems
[5] Estimation of crop yield from combined optical and SAR imagery using Gaussian kernel regression Authors: Alebele, Yeshanbele and Wang, Wenhui and Yu, Weiguo and Zhang, Xue	Journal: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	Estimation methods selected based on the tradeoff between the performance in terms of given target parameters, interpretability of results, and Computational time. Multiple linear regression, random forest, and neural networks are mostly used.	Using the Gaussian Linear regression model, a range of crop yield is predicted from an unknown distribution. GPR attempts to approximate the target output f(x) whereby interpreting it as a probability distribution function.	The method is sensitive to the choice of hyper-parameters. The method is not able to predict crop yield in real-time. The method is not able to generalize to new environments.	Developing a method for automatically selecting the optimal hyperparameters for the Gaussian kernel regression model. Developing a method for generalizing the model to new environments.

Chapter 4

Requirements

4.1 Hardware

Arduino Board: The central unit of your system. It controls the sensors, reads data, and displays information on the LCD.

DHT11 Sensor: This sensor measures temperature and humidity. It's useful for monitoring environmental conditions that affect plant growth.

LCD with I2C: An LCD screen with an I2C adapter allows you to display data conveniently. The I2C adapter reduces the number of pins required to connect the LCD to the Arduino, leaving more pins available for other components.

Breadboard: Provides a platform for connecting all the components without soldering. It makes prototyping and experimentation easier.

Jumper Wires: These are used to establish connections between the components on the breadboard and the Arduino.

Soil Moisture Sensor: Measures the moisture content of the soil. This data is crucial for determining when to water the plants, preventing overwatering or underwatering.

RTC Module (Real-Time Clock): Provides accurate timekeeping for the system. It's essential for scheduling tasks such as data logging or triggering actions at specific times.

4.2 Software

OpenweatherAPI - This was a vital resource for obtaining meteorological data needed to forecast appropriate crop yields. The API offers instant access to current meteorological information for any place on Earth, including temperature and humidity—two essential elements in judging a crop's adaptability. The required meteorological data was obtained, processed, and used in the prediction model to identify the best crop for the specified district by entering the district data into the API.

Visual studio code - Python scripting is needed to give the program dynamic and interactive behavior. Visual Studio Code is used to develop the scripts. The Flask framework was used in the creation of this application to create the user interface and connect it to the backend's trained machine learning models. Flask is a straightforward yet robust web framework that makes it possible to quickly prototype and launch online apps. Visual Studio Code, which offers an extensive toolkit for creating, testing, and debugging code, is being used to develop the application.

Google colab - Google's online platform, known as Google Colab, enables users to develop, execute, and collaborate on code within a Jupyter Notebook environment. The platform provides free usage of CPU, GPU, and TPU processing resources for tasks related to data analysis and machine learning. Colab offers an environment that is cloud-based. Colab also offers smooth connection with other Google services, such as Google Drive. All things considered, Colab is a useful and effective tool for machine learning and data science projects.

Chapter 5

Proposed System

5.1 System Architecture

Farmers face immense challenges in determining the ideal crop to cultivate for maximum profitability given a particular set of environmental conditions. Traditional approaches to crop forecasting have fallen short in providing robust predictive accuracy. To address this critical gap, we propose an integrated crop prediction, price estimation ,yield prediction and fertilizer prediction system leveraging both classification and regression models. The workflow is as follows: The farmer inputs key parameters into the application interface: soil nitrogen, phosphorus, and potassium levels, rainfall amount, and district name. The system retrieves the temperature and humidity for the location via the OpenWeatherMap API using the district name.In the main interface there are three models crop prediction,fertilizer recommendation and crop yield prediction when selected example crop prediction model The inputs are fed into the crop prediction model, which outputs the optimal crop for cultivation based on the conditions. The predicted crop and district name are input into the price estimation model, which forecasts expected monthly prices over the next 12 months. The crop recommendation and accompanying price data are presented to the farmer through the interface.when fertilizer model or crop yield model is selected after entering inputs for respective model the crop yield is given and fertilizer is recommended with the help of web interface .We've amalgamated a comprehensive dataset for crop management, blending resources from Kaggle for crop prediction with additional datasets sourced from Tamil Nadu University. These supplementary datasets encompass crop price prediction, fertilizer estimation, and crop yield analysis. This holistic approach ensures a robust foundation for informed decision-making in agricultural practices. The core innovation lies in the model architecture (Fig. 1). By sequentially connecting the classification and regression models, the system not only suggests the ideal crop but also provides actionable projections to maximize profits from the crop grown. This integrated approach provides robust predictive power from the synergistic use of AI techniques while delivering precise, customized insights to farmers. With higher accuracy and reliability than conventional methods, the proposed system has the potential to revolutionize data-driven decision support for the agricultural sector.

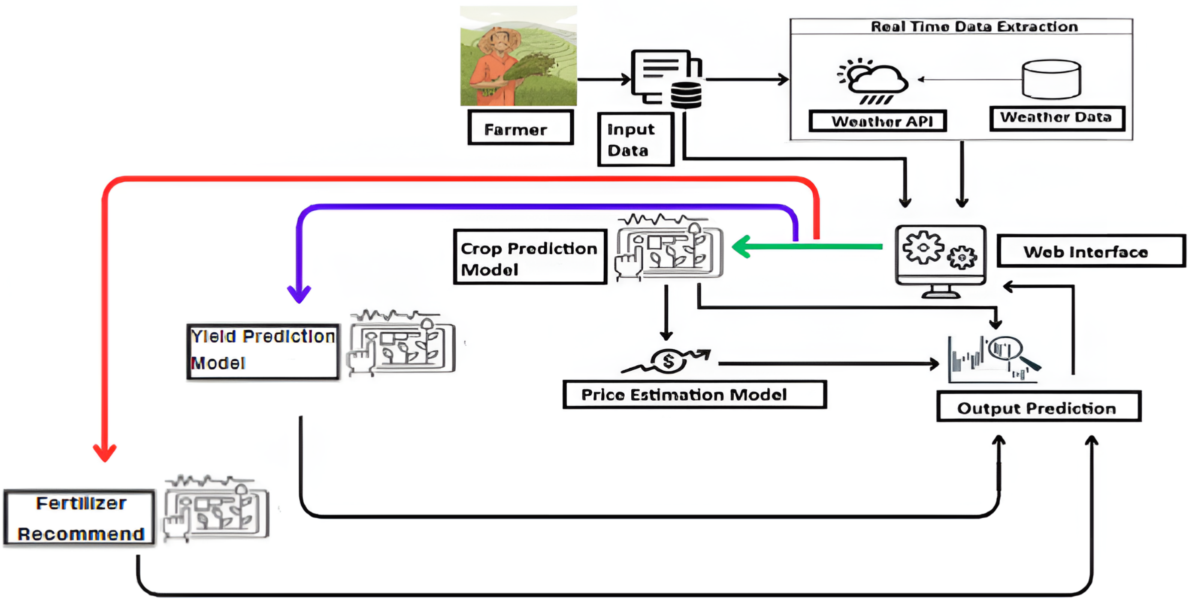


Figure 5.1: System Architecture

5.2 Crop Prediction Model

This module focuses on predicting the optimal crop for cultivation based on soil conditions and location within Tamil Nadu. The prediction relies on key independent variables: soil nitrogen, phosphorous, potassium levels, humidity, average rainfall, and temperature for the district. These inputs are sourced and synthesized from diverse datasets, including government portals like data.gov.in, indiastat.com, and the OpenWeatherMap API. The curated data is fed into a Crop Prediction Model designed as a voting ensemble classifier. To ensure reliable predictions, the raw data undergoes rigorous preprocessing: duplicate removal, irrelevant information filtering, missing value handling, value rounding, outlier elimination, and normalization/scaling. These steps enhance model convergence and performance.

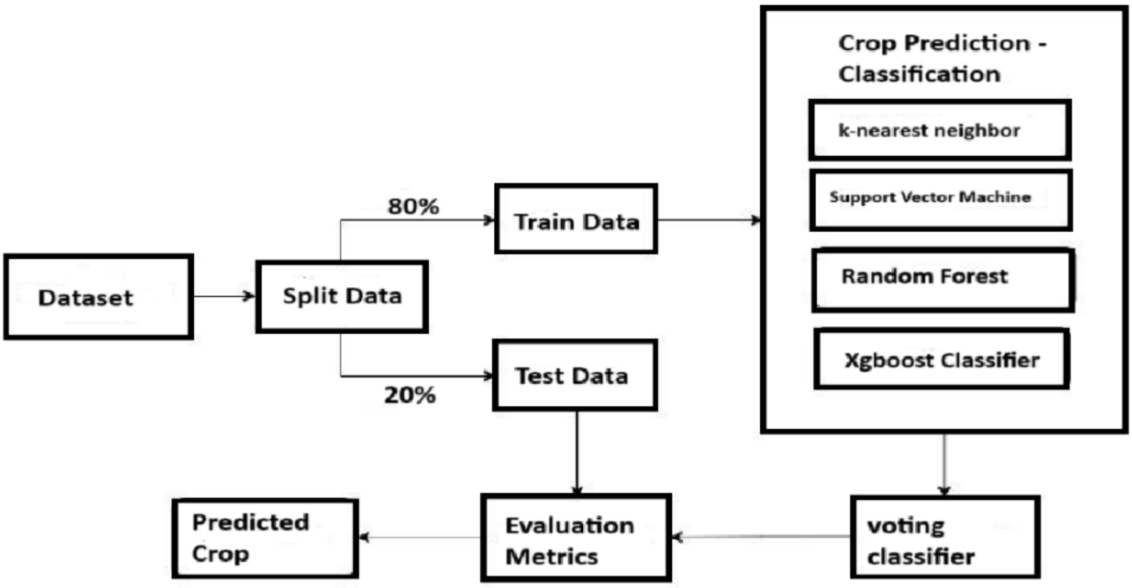


Figure 5.2: Crop Prediction Model

Algorithm 1 Crop Prediction Using Voting Classifier

Trained Crop Prediction classifiers: XGB_classifier, svm_classifier, knn_classifier, rnd_classifier Predicted Crop for the given test data sample
Initialize: estimators ← [XGB_classifier, svm_classifier, knn_classifier, rnd_classifier]
voting_classifier ← VotingClassifier(estimators, voting='hard')
Fit: voting_classifier.fit(X_train, y_train)
Predict: predicted_crop ← voting_classifier.predict(test_data_sample)
End

By aggregating predictions from an ensemble of diverse models, the system achieves higher accuracy and robustness compared to any individual model. The ensemble approach also safeguards against bias and overfitting. Overall, the system provides data-driven, location-specific crop recommendations to empower farmers with actionable insights for profit maximization.

5.3 Price Prediction Model

This module revolves around predicting crop prices for the subsequent 12 months in a specific district. The prediction process involves utilizing the crop name and district name as independent variables, which are derived from datasets extracted and synthesized from sources such as https://data.gov.in and www.indiastat.com. These datasets are input into the Price Estimation model, designed as a stacking regressor ensembler.

Algorithm 2 Crop Price Prediction Model

```
Input: Historical crop data D, Features X, Target y
Output: Predicted crop prices ŷ

procedure CropPricePrediction(D, X, y)
    Step 1: Preprocess data D
    Step 2: Split D into training and testing sets
    Step 3: Train base models
        ▷ Random Forest: RF ← RandomForestRegressor()
        ▷ Gradient Boosting: GB ← GradientBoostingRegressor()
        ▷ XGBoost: XGB ← XGBRegressor()
    Step 4: Stack base models using stacking regressor
        ▷ Stacker: stacker ← StackingRegressor(estimators = [RF, GB, XGB])
    Step 5: Train the stacker on the training data
        stacker.fit(X_train, y_train)
    Step 6: Predict crop prices on the testing set
        ŷ ← stacker.predict(X_test)
    Step 7: Evaluate the model performance
        ▷ Used MSE, MAE and RMSE evaluation metrics.
    Step 8: Return predicted crop prices ŷ
end procedure
```

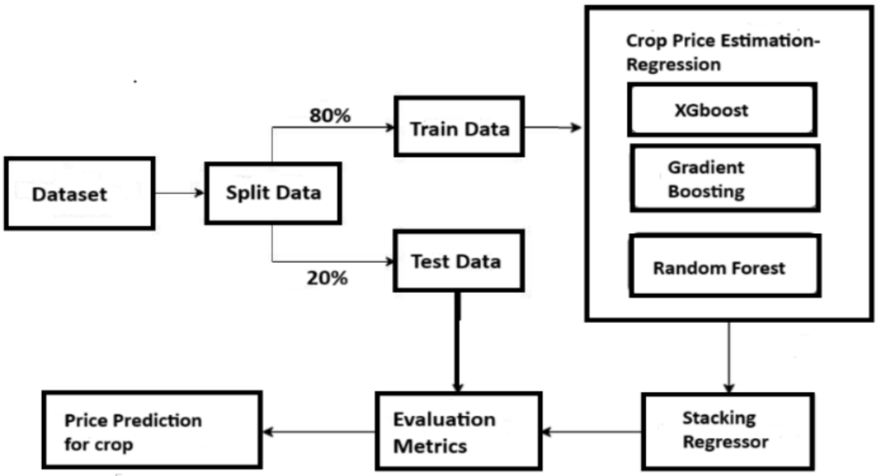


Figure 5.3: Price Prediction Model

To ensure the accuracy of the predictions, the collected data undergoes a meticulous cleaning process. This involves the removal of duplicates, filtering out irrelevant data, handling missing values, rounding values, and eliminating outliers. Encoding is carried out using Label Encoder for the labeled classes. The regression models employed in this context include XGBoost regression, Random Forest regression, and Gradient boosting. Figure 5.5 illustrates the Price Prediction Model.

5.4 Crop Yield Prediction Model

Using three machine learning models—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest—provides an extensive investigation of predictive powers in the field of crop yield prediction.Following a comprehensive research, the Random Forest model emerged as the best option due to its resilience and versatility. The Random Forest ensemble learning approach, which integrates numerous decision trees, was critical for decreasing overfitting and improving prediction accuracy. Random Forest's non-parametric qualities and capacity to judge feature importance make it extremely good in identifying detailed patterns in agricultural data, increasing the accuracy of production estimations.

Random Forest beats SVM and KNN for a number of reasons. Its resistance to overfitting is a key benefit, especially when working with noisy and complicated agricultural datasets. The model's scalability enables it to easily handle big agricultural datasets, unlike SVM, which may struggle with computing demands. Furthermore, Random Forest has intrinsic interpretability and the ability to prioritise feature significance, which provides useful insights into agricultural yield patterns. This provides stakeholders with actionable intelligence so they may make educated agricultural management decisions.

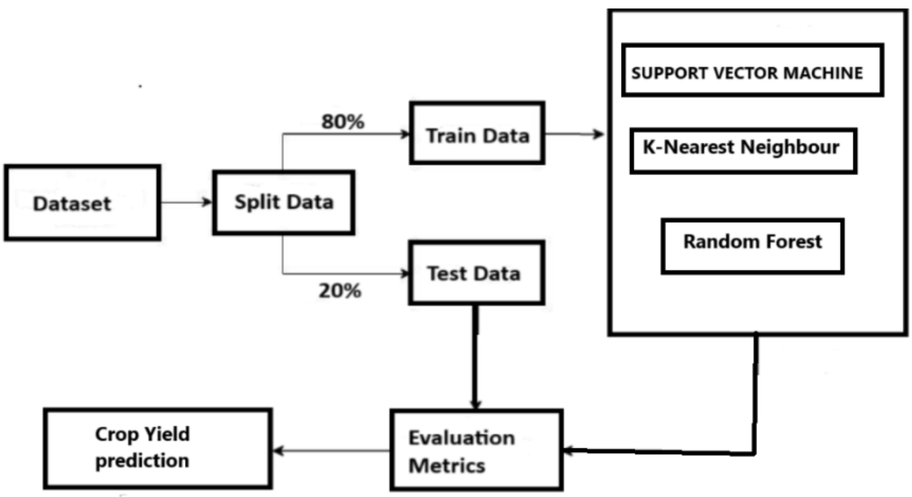


Figure 5.4: Yield prediction Model

Algorithm 3 Crop Yield Prediction Model

Input: Historical crop data D, Features X, Target y
Output: Predicted crop yield $\hat{y}$

procedure CropYieldPrediction(D, X, y)
    Step 1: Preprocess data D
    Step 2: Split D into training and testing sets
    Step 3: Train base models
        ▷ Random Forest: RF ← RandomForestRegressor()
        ▷ Support Vector Machine (SVM): SVM ← SVR()
        ▷ k-Nearest Neighbors (k-NN): kNN ← KNeighborsRegressor()
    Step 4: Evaluate base models
        ▷ Used MSE, MAE and RMSE evaluation metrics.
    Step 5: Select the best model
        ▷ Choose the model with the best performance (e.g., Random Forest)
    Step 6: Train the selected model on the full training data
        best_model ← RandomForestRegressor()   ▷ As Random Forest is selected
        best_model.fit(X, y)
    Step 7: Predict crop yield on the testing set
        $\hat{y}$ ← best_model.predict(X_{test})
    Step 8: Return predicted crop yield $\hat{y}$
end procedure

It is critical to continue improving and analysing the Random Forest model to guarantee that it appropriately predicts crop yields. Experimenting with hyperparameters and feature engineering can assist improve accuracy. Furthermore, matching the model's projections with agricultural experience promotes trust and pragmatism, helping farmers, policymakers, and stakeholders to make better decisions. As the agricultural environment changes, harnessing Random Forest's predictive skills can create new options for enhancing crop management and guaranteeing food security.

5.5 Fertilizer Prediction Model

It is critical to use the appropriate machine learning algorithm for anticipating fertiliser needs in order to obtain accurate and efficient outcomes. This entails carefully weighing many possibilities, each of which has advantages and disadvantages.

SVMs are excellent at managing vast and complicated datasets, making them ideal for fertiliser prediction applications. Random Forest, on the other hand, employs a network of decision trees to provide a dependable solution for dealing with large datasets while minimising the danger of overfitting. Furthermore, XGBoost, a cutting-edge gradient boosting algorithm, excels at both performance and scalability, making it ideal for applications that need great precision and efficiency.

Among these methods, the Bagging Classifier comes out as the best solution for fertiliser prediction. This approach takes use of ensemble learning by merging many models, each of which focuses on a different region of the dataset. The Bagging Classifier frequently generates more accurate and comprehensive results than individual algorithms by combining information from these models via bootstrapping and aggregation. This collaborative approach not only improves prediction accuracy but also increases the model's robustness to the dataset's complexity and biases.

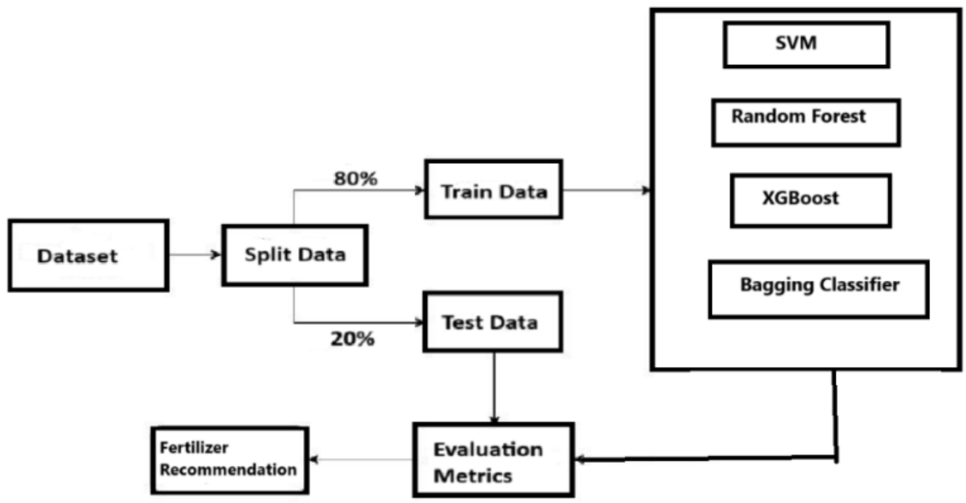


Figure 5.5: Fertilizer prediction Model

The main advantage of the Bagging Classifier in estimating fertiliser requirements emphasises the value of ensemble approaches for complicated jobs. By using the combined wisdom of numerous models, the Bagging Classifier overcomes the limits of individual algorithms, resulting in a complete and robust fertiliser recommendation solution. Its capacity to extract insights from multiple portions of the information enhances forecast accuracy while also encouraging a deeper knowledge of the patterns that determine fertiliser requirements. As such, Bagging Classifier emerges as a beacon of innovation in the realm of agricultural analytics, empowering farmers and stakeholders with actionable insights for sustainable crop management and optimal resource utilization.

Algorithm 4 Fertilizer Prediction Model with Evaluation Metrics

```
Input: Historical fertilizer data D, Features X, Target y
Output: Predicted fertilizer ŷ

procedure FertilizerPrediction(D, X, y)
    Step 1: Preprocess data D
    Step 2: Split D into training and testing sets
    Step 3: Train base models
        ▷ Support Vector Machine (SVM): SVM ← SVC()
        ▷ Random Forest: RF ← RandomForestClassifier()
        ▷ XGBoost: XGB ← XGBClassifier()
    Step 4: Train Bagging Classifier
        ▷ Bagging: bagging ← BaggingClassifier(base_estimator = [SVM, RF, XGB])
    Step 5: Train the Bagging Classifier on the training data
        bagging.fit(X_train, y_train)
    Step 6: Predict fertilizer on the testing set
        ŷ ← bagging.predict(X_test)
    Step 7: Evaluate the model
        ▷ Compute accuracy, precision, F1 score, and recall
    Step 8: Return predicted fertilizer ŷ
end procedure
```

5.6 Web Interface

The user interface of the web application is designed with the farmer's convenience in mind, facilitating the effortless input of data and the retrieval of predictions regarding the most suitable crop for cultivation in their specific location. Constructed using a combination of tools, including Python Virtual Environment, Flask, HTML/CSS, and Bootstrap, the application ensures a user-friendly and visually appealing interface that is highly customizable.

The input page of the application is meticulously crafted to gather crucial information from farmers, including Nitrogen, Phosphorous, Potassium (NPK) levels, District details, Rainfall, and a submit button to finalize the input form. This input data serves as input for the trained machine learning models running in the backend, enabling accurate predictions about the optimal crop,yield for the farmer to cultivate.

Upon submission of the input data, the application navigates the farmer to the prediction page, which comprises two key sections. The first section features a price prediction table for the forecasted crop, displaying the anticipated prices over the next 12 months. The second section incorporates a chart or graph visualizing the projected prices, providing farmers with an intuitive way to interpret the data and make well-informed decisions on when to sell their crops.By accessing the fertilizer or yield section in the web interface and entering the necessary input values, you will obtain the desired results.

5.7 Crop monitoring system

Crop monitoring systems include technologies such as Arduino, DHT11 sensors, I2C LCDs, soil moisture sensors, and RTC modules to improve agricultural efficiency. The soil moisture sensor checks soil wetness, while the DHT11 sensor monitors temperature and humidity. Data from these sensors is presented on an LCD screen via I2C communication, giving farmers with critical real-time information. This enables them to make better judgements regarding irrigation, pest management, and other agricultural operations, resulting in improved crop health and yields across a variety of crops and farming practices.

The RTC module ensures precise timekeeping while also allowing for work scheduling and data reporting. Using exact timestamps, farmers may automate chores such as collecting data at regular intervals and operating irrigation equipment at appropriate periods. This automation minimises the need for physical labour, freeing farmers to concentrate on more important choices and resource management. Farmers may also customise the system with Arduino to match their individual needs, making it adaptable to a wide range of crops and farming approaches.

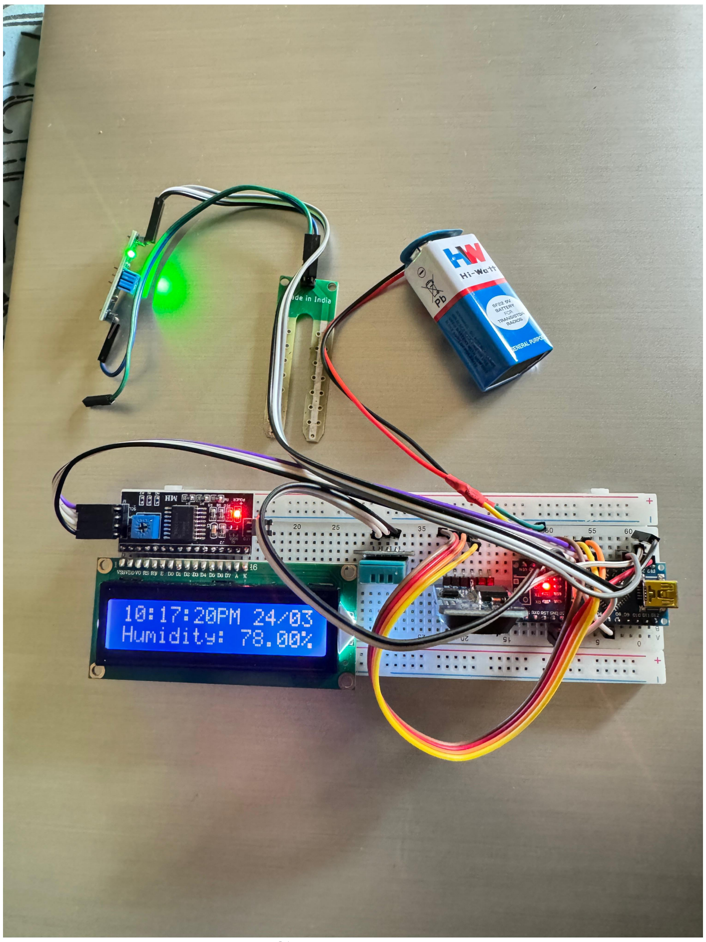


Figure 5.6: Crop monitoring system

The Arduino agricultural monitoring system takes a complete approach to farm management. Sustainability and productivity are improved by combining automation, data visualisation, and sensors. Farmers may boost crop yields by using accurate timing and real-time environmental data, while minimising resource use and environmental effect. As agriculture faces difficulties like as climate change and increased food demand, using modern monitoring systems is critical to guaranteeing farming's long-term sustainability and resilience.

Chapter 6

Result and Analysis

6.1 Result 1 - Crop Prediction

The proposed models for crop prediction and estimate have been thoroughly tested in a variety of contexts, producing validated results. This method uses four crop prediction models: random forest classifier, Support Vector Machine, XGBOOST, and KNN classifier. When evaluating the performance metrics of a classification model, we often start by creating a confusion matrix using the model's predictions and the actual labels. The confusion matrix includes True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).From these metrics, we can calculate several evaluation metrics. Accuracy, representing the proportion of correctly classified samples among the total number of samples, is computed using the formula $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. Precision, which measures the proportion of correctly predicted positive samples among all samples predicted as positive, is calculated as $Precision = \frac{TP}{TP+FP}$. Recall, also known as sensitivity, quantifies the proportion of correctly predicted positive samples among all actual positive samples, and is computed as $Recall = \frac{TP}{TP+FN}$. Finally, the F1 Score, which balances precision and recall, is computed as the harmonic mean of precision and recall, given by $F1Score = \frac{2 \times Precision \times Recall}{Precision+Recall}$. It is based on a Voting Classifier. The accuracy metrics of the four models that the voting classifier is utilizing As shown in Table 6.1.

Table 6.1: Crop prediction performance metrics

Model	F1 Score	Accuracy	Recall	Precision
Random Forest Classifier	0.978	0.983	0.978	0.979
Support Vector Machine	0.959	0.978	0.957	0.954
KNN Classifier	0.976	0.967	0.976	0.975
XGBoost	0.978	0.983	0.978	0.983

Voting was used to assess each model's performance, and the XGBoost Classifier produced the best results in terms of precision. Figure 6.2, the confusion matrix showing the quantity of false positives, false negatives, true positives, and true negatives, provides an illustration of this.

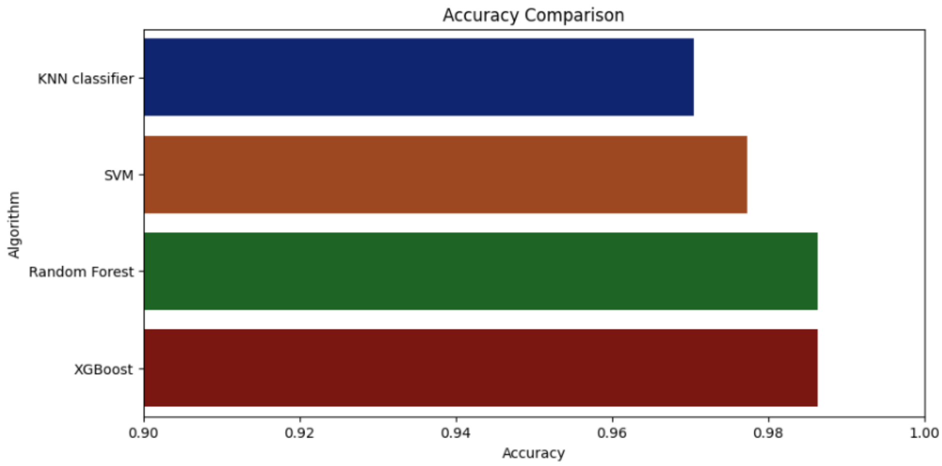
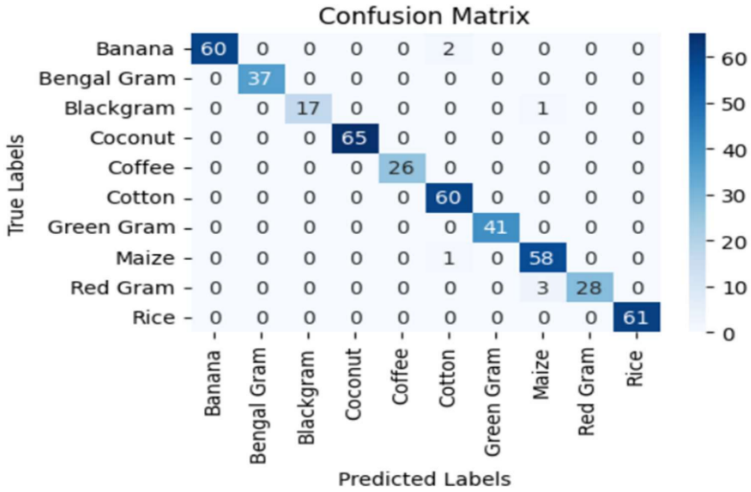


Figure 6.1: Accuracy Comparison

The findings show that the Voting Classifier chooses the most advantageous model with the most votes to correctly predict the right crop to plant after merging a variety of models. When compared to the individual models, it exhibits a significant improvement with a highest accuracy of 0.987. This illustrates how well the XG BOOST performs in selecting the best crop given the available data.




```
Overall accuracy: 0.9847826086956522
Overall precision: 0.9887864823348694
Overall recall: 0.9798463033837554
Overall F1-score: 0.9838475113226476
```

Figure 6.2: Voting Classifier Confusion Matrix

6.2 Result 2 - Price prediction

Table 6.3 shows the performance of the different regression models which are trained for Crop Price Estimation. When evaluating regression models, Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) are commonly used metrics. MSE measures the average squared difference between the predicted values and the actual values, computed as $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where y_i represents the true values and $\hat{y}_i$ represents the predicted values. RMSE, which is the square root of MSE, provides a measure of the average magnitude of the errors, and is calculated as $RMSE = \sqrt{MSE}$. MAE measures the average absolute difference between the predicted values and the actual values, given by $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$. These metrics help assess the accuracy and generalization capability of regression models.

Table 6.2: Price estimation performance metrics

Model	MSE	RMSE	MAE
Gradient Boosting Regressor	380.71	19.51	16.52
XG Boosting Regressor	357.61	18.91	16.10
Random Forest Regressor	366.81	19.015	16.28

The selected ideal hyper parameters along with the associated error values. These parameters learning rate,max depth,n-estimators, reg-lambda are analyzed to determine the models correctness and efficiency in pricing estimation. The image as shown in Figure 6.3 clearly illustrates the great efficiency of the stacking regressor, which takes advantage of the variety of base models to capture various features of the data. This shows that the stacking regressor improves the overall performance of the price estimation system by effectively combining the advantages of several models to generate strong and accurate price estimates. The model is trained when the optimal parameters are identified through hyperparameter tuning.

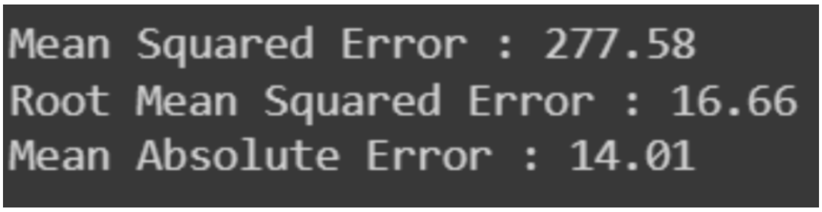


Figure 6.3: Stacking Regressor Evaluation Metrics

For price estimation, the model is based on a Stacking Regressor with the top 2 models - Random Forest Regressor and XGBoosting Regressor. Figure 6.3 shows that the Stacking Regressor has an optimized Mean Absolute Error of 14.01 which is best compared to the individual Regression models. Thus, implying this ensemble model fits the input data much better. Stacking Regressor is a machine learning ensemble method that combines multiple regression models to improve prediction accuracy. It operates by training several base regression models on the same dataset and then using a meta-regressor to learn from the predictions of these base models. The meta-regressor uses predictions from various base models to get the final forecast. The Stacking Regressor takes advantage of the many perspectives provided by each base model to improve comprehension of the data and eliminate biases inherent in the individual models. It usually surpasses any single model since it combines the strengths of several models. The Stacking Regressor is a flexible and efficient regression approach that improves prediction accuracy and dependability.

6.3 Result 3 - Crop yield prediction

Crop yields may be predicted using a variety of approaches, including Support Vector Machine (SVM), k-Nearest Neighbours (KNN), and Random Forest. Random Forest outperforms the other methods because it excels at properly projecting yields by diving into detailed patterns in the data. This strategy combines many decision trees, which not only decreases mistakes but also increases reliability. Random Forest is commonly used by farmers and policymakers to produce reliable agricultural predictions due to its high accuracy.

Table 6.3: Crop yield performance metrics

Model	MAE	MSE	RMSE
KNN	21.468	761.267	27.591
SVM	21.783	788.780	28.085
Random Forest Regressor	19.8405	646.359	25.42

On the contrary, SVM and KNN excel in predicting agricultural yields, but their effectiveness is dependent on the precise data and parameters used. SVM excels at determining appropriate borders to separate distinct data classes, but it may struggle with large datasets or complex connections. KNN, on the other hand, is simple and understandable, however its results are strongly influenced by the distance measuring method and the number of neighbours used. Both SVM and KNN provide alternate ways to forecasting crop yields, providing to a wide range of agricultural scenarios and needs.

Choosing between SVM, KNN, and Random Forest for agricultural production prediction depends on various criteria, including data complexity, available computational resources, and the need for understandable findings. While Random Forest produces extremely accurate results, SVM and KNN can provide simplicity and versatility in specific situations. As a result, recognising the strengths and limits of each strategy is critical for picking the best one to improve farming efficiency and resource management efficiently.After models evaluation we came to conclusion that random forest is the best because its scores are far better than the other two models which are SVM and KNN.

6.4 Result 4 - Fertilizer prediction

Fertiliser plays a crucial part when it comes to appropriate farming methods. Predicting fertiliser is critical in modern agricultural techniques, since it optimises resource allocation and increases crop output. Several approaches, including Random Forest, XGBoost, Bagging Classifier, and Support Vector Machines (SVM), have substantial capabilities for this job.Random Forest is an established ensemble approach that builds many decision trees and aggregates their forecasts to improve accuracy and dependability. XGBoost, a revised gradient boosting framework, systematically constructs decision trees to correct flaws in previous models, displaying exceptional speed and efficacy, especially with structured data.The Bagging Classifier uses bootstrap sampling to train several base models on different data subsets, resulting in reduced variance and increased stability.Notably, the Bagging Classifier stands out for its capacity to generate reliable forecasts while reducing the danger of overfitting, making it an appropriate candidate for fertiliser prediction, particularly in the face of noisy and complicated data.

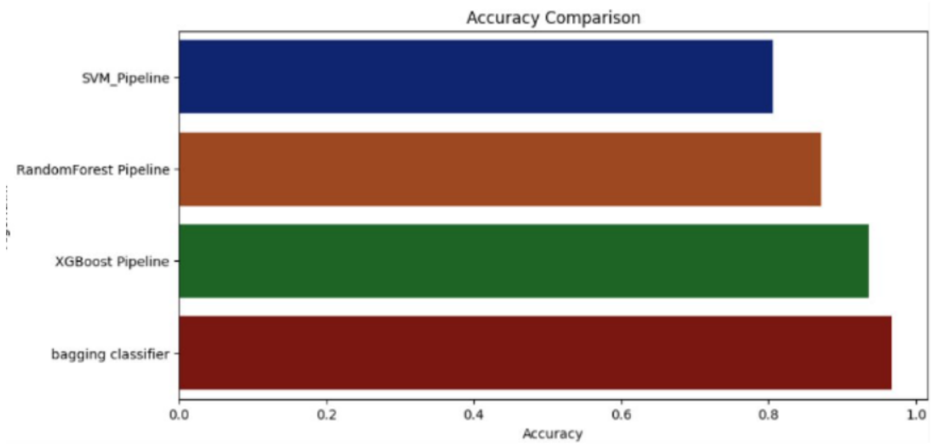


Figure 6.4: Accuracy Comparison Graph

In agricultural situations, the Bagging Classifier appears as a dependable and robust alternative, particularly capable of managing diverse and noisy data. The Bagging Classifier reduces overfitting and enhances generalisation performance by combining predictions from numerous base models trained on bootstrapped data subsets. This capability is extremely useful in fertiliser prediction jobs, where complex correlations between input variables and fertiliser requirements emerge as a result of variations in soil composition, meteorological conditions, and crop varieties. Furthermore, the collaborative nature of the Bagging Classifier assures effective collection of numerous patterns in data, resulting in more accurate predictions than solo classifiers.

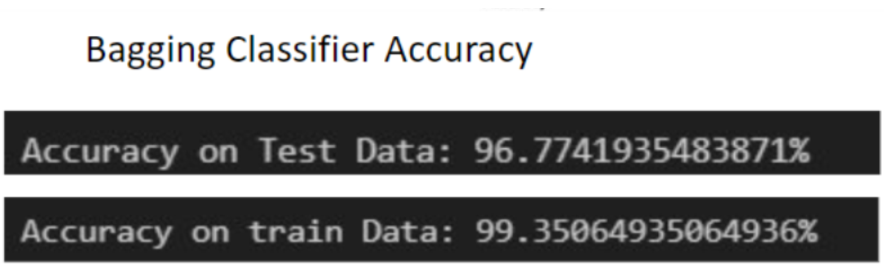


Figure 6.5: Bagging classifier Accuracy

The Bagging Classifier's ability to handle large datasets contributes to its use as a fertiliser prediction tool. Scalability is crucial in agriculture since datasets usually span wide geographical and temporal boundaries. The Bagging Classifier excels at processing big agricultural datasets because to its parallelism and computational efficiency. This enables it to make more timely and accurate fertiliser recommendations for a variety of environmental conditions.Furthermore, its adaptability enables it to accept a wide range of input components, including soil factors, crop quality, and meteorological conditions. Because of its versatility, the Bagging Classifier is an efficient tool for dealing with contemporary farming challenges and enhancing crop productivity.

6.5 Result 5 - Web Interface

The primary interface of the web system comprises three main modules, as illustrated in Figure 6.6. Upon selecting the first module, the user is directed to the form depicted in Figure 6.7, situated within the application interface. This form facilitates input of N, P, K ratios, the district's name, and rainfall (in mm) from the user. All fields in the form are mandatory

Figure 6.6: Web application

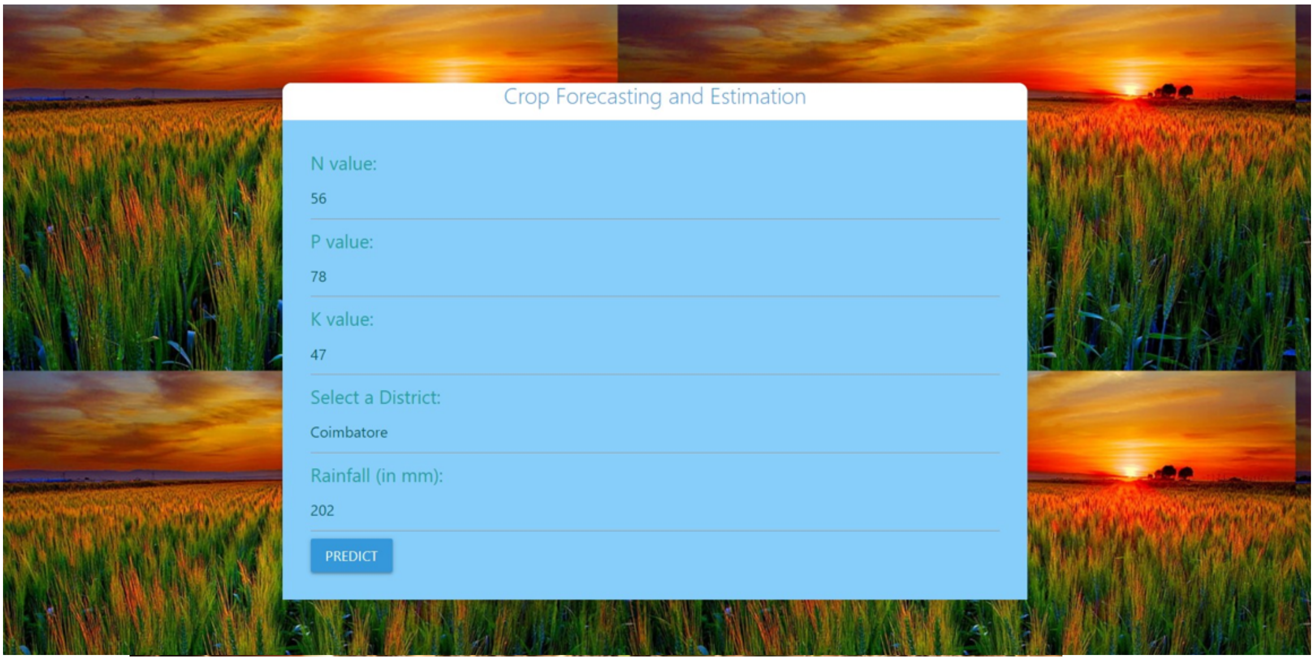


Figure 6.7: User Input page

The data from the application is passed into the first model – crop prediction model. This model predicts the most suitable crop for the given climatic and soil conditions. This value along with the district name is fed as input to the price estimation model which now predicts the crop prices for the next 12 months. Figure 6.8 shows the crop and the corresponding list of prices in tabular format. The web application design is done using HTML, CSS, and Flask UI, and the trained model is imported in the web UI as a .pkl file.

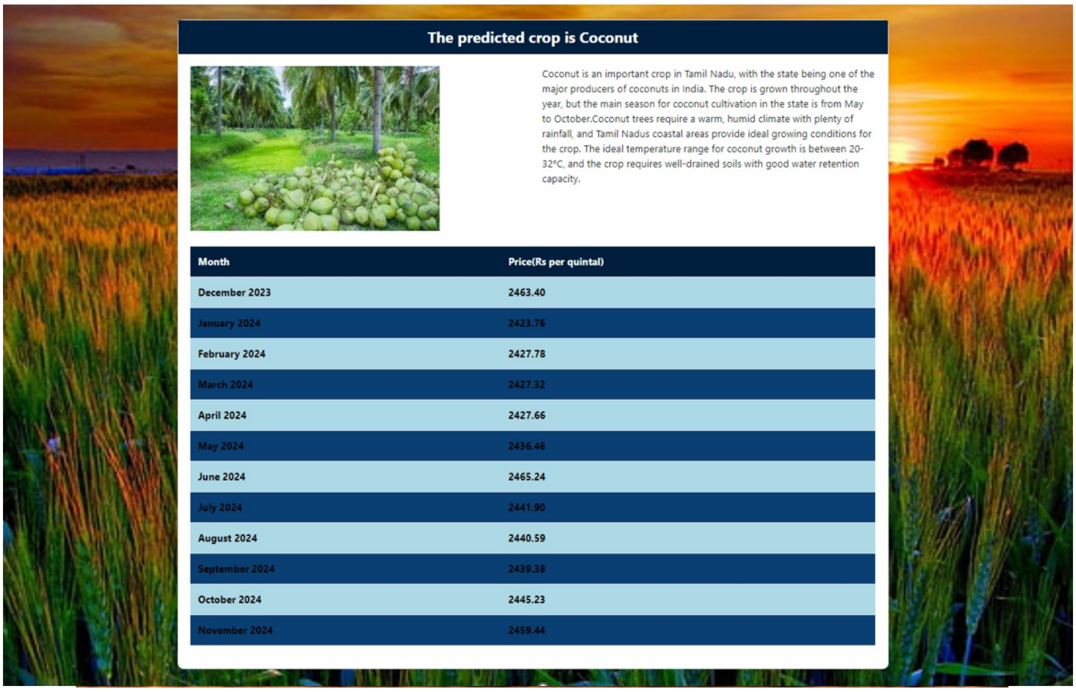


Figure 6.8: Predicted crop with price

The UI further shows the price trend as a graph for the farmer to better understand the best time for cultivation.

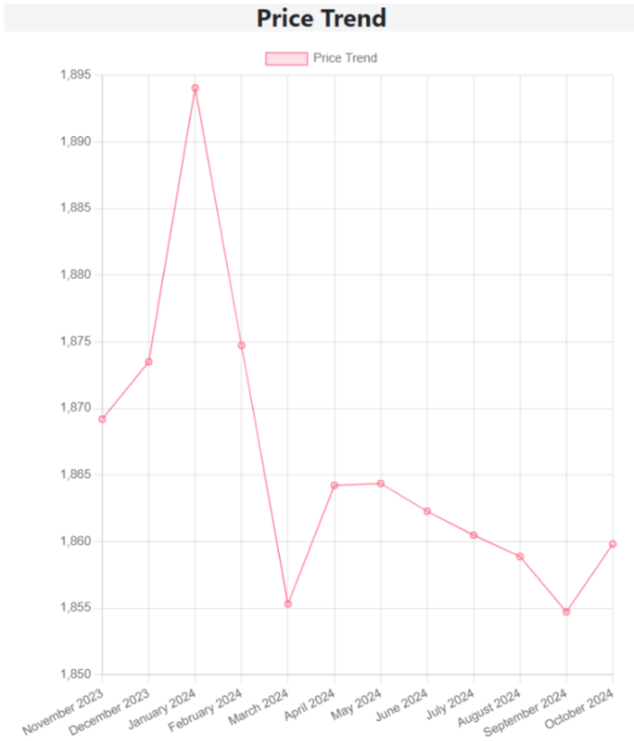


Figure 6.9: Price Trends Line Graph

The Agricultural Prediction System offers two other distinct functionalities to assist farmers in optimizing their crop management strategies. upon selecting fertilizer model we get web interface as shown in figure 6.10, farmers can utilize the Fertilizer Prediction module by inputting essential parameters such as soil type, crop type, soil pH, nutrient levels, and climate conditions. Using this data, the system generates recommendations for the most suitable fertilizer, tailored to the specific needs of the crop and soil.

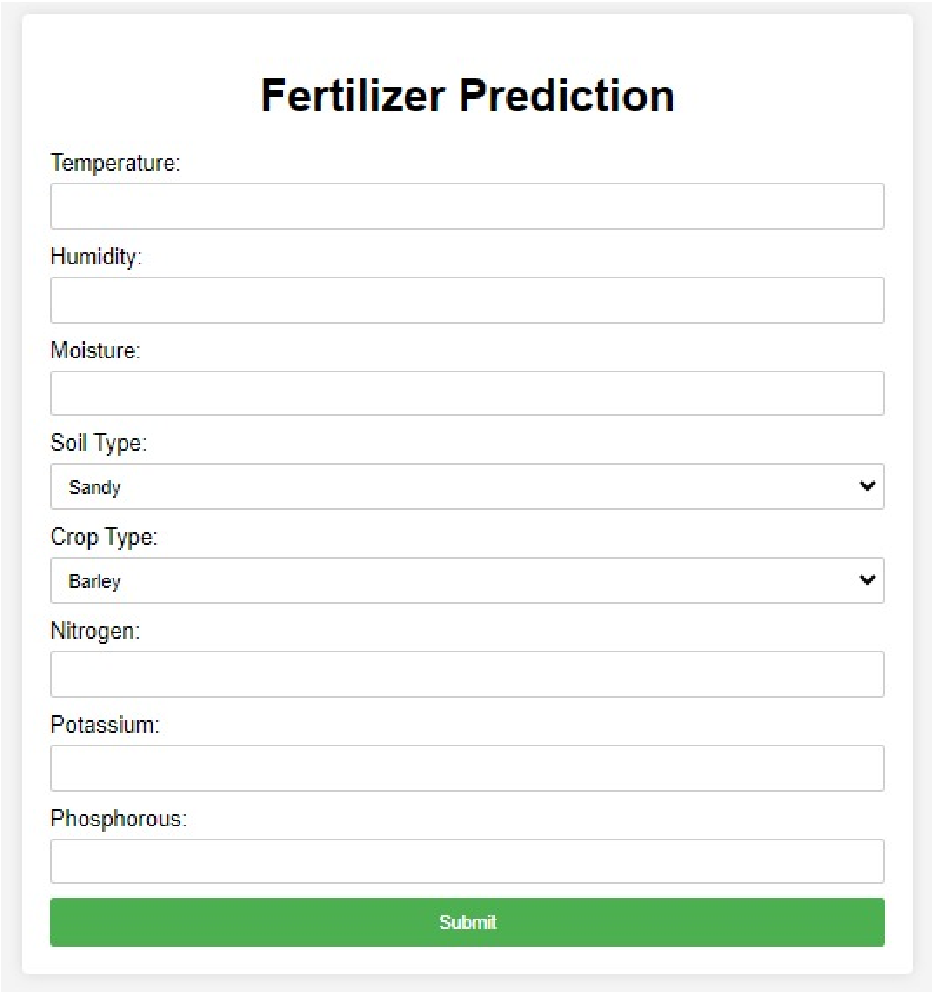


Figure 6.10: Fertilizer prediction webpage

Secondly upon clicking yield model as shown in Figure 6.11, the system provides a Yield Prediction feature, where farmers input parameters including crop type, soil type, climate conditions, irrigation practices, pest and disease management, and previous crop yield data if available. Leveraging this information, the system predicts the expected yield in tonnes per hectare, aiding farmers in planning and decision-making for optimal agricultural productivity. Together, these modules empower farmers with actionable insights to enhance their farming practices and improve overall crop yields.

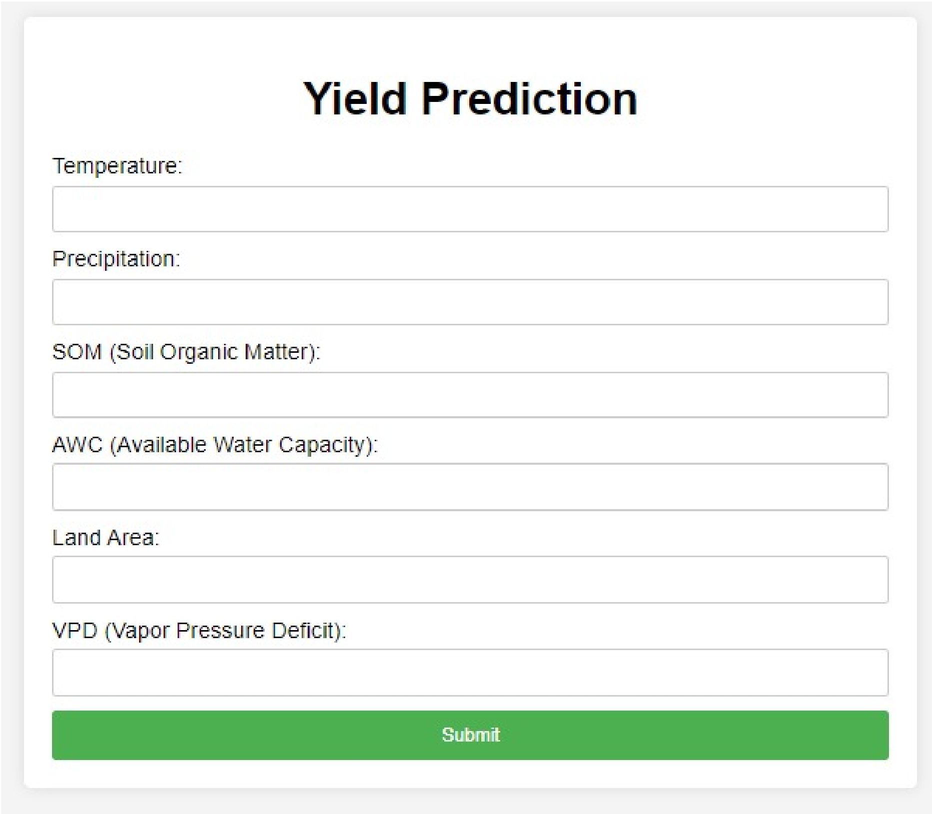


Figure 6.11: yield prediction webpage

Chapter 7

Conclusion

7.1 Conclusions

The prediction system being proposed provides a solution to the many difficulties experienced by farmers in different areas of agriculture, such as crop selection, price prediction, fertilizer suggestions, and yield estimation. By utilizing advanced machine learning methods and effectively incorporating essential weather and soil information, this ground-breaking solution can accurately identify the best crop for any specific area. It also estimates agricultural commodity prices for the coming year and offers assistance on estimating crop yields and optimising fertiliser use. The excellent performance of our machine learning algorithms, which were meticulously constructed and thoroughly tested on real-world data, indisputably indicates the system's use. These practical findings show how this technology has the potential to alter farming practices throughout the world.In essence, the system has been carefully built to offer farmers with actionable data to assist them in crop selection and marketing decisions. These crucial insights have the potential to boost agricultural productivity and profitability for farmers. With its integrated and research-backed structure, this solution emerges as a light of hope for the agriculture industry, offering improved prospects and more financial success for agricultural stakeholders in the future.

7.2 Future Work

In the future, we hope to take sensor readings directly within a web page utilising a cloud platform, allowing for real-time data capture and presentation via technologies such as JavaScript and WebSocket. To improve speed for bigger datasets, we want to optimise database design, incorporate caching methods, and use scalable cloud infrastructure. In addition, we want to improve the website by adding features like user authentication, alerting systems, data analytics, and user customisation choices. Furthermore, adding external APIs will increase functionality by delivering new services such as weather predictions or geographic data, improving the overall user experience.

References

[1] D. Elavarasan and P. M. D. Vincent, "Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications," in *IEEE Access*, vol. 8, pp. 86886-86901, 2020, doi: 10.1109/ACCESS.2020.2992480.

[2] Shedthi, B. Shabari, et al. "Crop and Nutrient Recommendation System using Machine Learning for Precision Agriculture." 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE). IEEE, 2022.

[3] Priyadharshini, A., et al. "Intelligent crop recommendation system using machine learning." 2021 5th international conference on computing methodologies and communication (ICCMC). IEEE, 2021.

[4] Jaichandran, R., et al. "Light GBM Algorithm based Crop Recommendation by Weather Detection and Acquired Soil Nutrients." 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS). IEEE, 2022.

[5] Alebele, Yeshanbele, et al. "Estimation of crop yield from combined optical and SAR imagery using Gaussian kernel regression." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021): 10520-10534.

[6] F. Ji, J. Meng, Z. Cheng, H. Fang and Y. Wang, "Crop Yield Estimation at Field Scales by Assimilating Time Series of Sentinel-2 Data Into a Modified CASA-WOFOST Coupled Model," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-14, 2022, Art no. 4400914, doi: 10.1109/TGRS.2020.3047102.

[7] S. Vashisht, P. Kumar and M. C. Trivedi, "Improvised Extreme Learning Machine for Crop Yield Prediction," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 754-757, doi: 10.1109/ICIEM54221.2022.9853054.

[8] S. M. M. Nejad, D. Abbasi-Moghadam, A. Sharifi, N. Farmonov, K. Amankulova and M. Lászlź, "Multispectral Crop Yield Prediction Using 3D-Convolutional Neural Networks and Attention Convolutional LSTM Approaches," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 254-266, 2023, doi: 10.1109/JSTARS.2022.3223423.

[9] M. Qiao et al., "Exploiting Hierarchical Features for Crop Yield Prediction Based on 3-D Convolutional Neural Networks and Multikernel Gaussian Process," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4476-4489, 2021, doi: 10.1109/JSTARS.2021.3073149.

[10] G. Mariammal, A. Suruliandi, S. P. Raja and E. Poongothai, "Prediction of Land Suitability for Crop Cultivation Based on Soil and Environmental Characteristics Using Modified Recursive Feature Elimination Technique With Various Classifiers," in IEEE Transactions on Computational Social Systems,vol. 8, no. 5, pp. 1132-1142, Oct. 2021, doi: 10.1109/TCSS.2021.3074534.

[11] Priya, S. Shanmuga, and R. Adwait Dathan. "Attention based Peephole LSTM model for Soybean crop yield prediction." Journal of Physics: Conference Series. Vol. 2571. No. 1. IOP Publishing, 2023.

[12] V. Valsan, A. M. A. Sai, A. R. Devidas and M. V. Ramesh, "Regression based Prediction of Rainfall for Energy Management in a Rural Islanded Micro-Hydro Grid in Kerala," 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), New Delhi, India, 2022, pp. 1-7, doi: 10.1109/GlobConPT57482.2022.9938353.

[13] P. Rekha, V. P. Rangan, M. V. Ramesh and K. V. Nibi, "High yield groundnut agronomy: An IoT based precision farming framework," 2017 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, USA, 2017, pp. 1-5, doi: 10.1109/GHTC.2017.8239287.

[14] Varghese, Arpitha, and I. Mamatha. "A Unified System for Crop Yield Prediction, Crop Recommendation, and Crop Disease Detection." International Conference on Robotics, Control, Automation and Artificial Intelligence. Singapore: Springer Nature Singapore, 2022.

[15] S. M. Kuriakose and T. Singh, "Indian Crop Yield Prediction using LSTM Deep Learning Networks," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-5, doi: 10.1109/ICCCNT54827.2022.9984407.

[16] Valsan, Vipina, et al. "Regression based Prediction of Rainfall for Energy Management in a Rural Islanded Micro-Hydro Grid in Kerala." 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT). IEEE, 2022.

[17] T. M. Swetha, T. Yogitha, M. K. Sai Hitha, P. Syamanthika, S. S. Poorna and K. Anuraj, "IOT Based Water Management System For Crops Using Conventional Machine Learning Techniques," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-4, doi: 10.1109/ICCCNT51525.2021.9579651.

Appendix A

A.1 Dataset

To guarantee data quality, a series of data preprocessing techniques were implemented to all the datasets of crop prediction ,crop price estimation,crop yield prediction and fertilizer recommendation comprising the elimination of duplicate entries, exclusion of extraneous information, imputation of lacking values, rounding off of figures, and resolution of anomalous observations. Additionally, encoding was performed using a Label Encoder for the labeled classes.

A.1.1 Crop Prediction

This dataset contains information on soil characteristics, rainfall, and temperature in 22 different crops grown in India. The dataset also includes information on the crops that are grown at different temperatures. The dataset as shown in fig A.1 has 6 attributes.Rainfall:The average annual rainfall in the location. Temperature:Temperature of the location in degrees Celsius PH:The pH of the soil in the location. Nitrogen (N), Phosphorus (P), Potassium (K):The amount of N, P, K in the soil in the location.

N	P	K	temperatu	humidity	rainfall	Crop
90	42	43	20.87974	82.00274	202.9355	rice
85	58	41	21.77046	80.31964	226.6555	rice
60	55	44	23.00446	82.32076	263.9642	rice
74	35	40	26.4911	80.15836	242.864	rice
78	42	42	20.13017	81.60487	262.7173	rice
69	37	42	23.05805	83.37012	251.055	rice
69	55	38	22.70884	82.63941	271.3249	rice
94	53	40	20.27774	82.89409	241.9742	rice
89	54	38	24.51588	83.53522	230.4462	rice
68	58	38	23.22397	83.03323	221.2092	rice
91	53	40	26.52724	81.41754	264.6149	rice
90	46	42	23.97898	81.45062	250.0832	rice
78	58	44	26.8008	80.88685	284.4365	rice
93	56	36	24.01498	82.05687	185.2773	rice
94	50	37	25.66585	80.66385	209.587	rice
60	48	39	24.28209	80.30026	231.0863	rice
85	38	41	21.58712	82.78837	276.6552	rice
91	35	39	23.79392	80.41818	206.2612	rice
77	38	36	21.86525	80.1923	224.555	rice

Figure A.1: Crop prediction dataset

The dataset contains 22 different crops labels - Apple, banana, black gram, chick-pea, coffee, cotton, grapes, jute, kidney beans, lentil, maize, mango, muskmelon, moth beans, orange, papaya, pigeon peas, pomegranate, rice, watermelon,corn.

A.1.2 Crop Price Estimation

This dataset as shown in fig A.2 contains information on crops grown in different districts of Tamil Nadu state and sold. There are a total of 32 districts and 22 different crops for each district and the time of crop sold from Jan - 2001 to Dec - 2020. The price of crop sold per quintal.Features - District, Crop Grown, Date (Sold), Crop Price (Per Quintal)

District	Crop	Price Date	Crop Price (Rs per quintal)
Ariyalur	Banana	Jan-15	1830
Ariyalur	Banana	Jan-15	1820
Ariyalur	Banana	Feb-15	1730
Ariyalur	Banana	Mar-15	1780
Ariyalur	Banana	Apr-15	1900
Ariyalur	Banana	May-15	1850
Ariyalur	Banana	Jun-15	1845
Ariyalur	Banana	Jun-15	1875
Ariyalur	Banana	Jul-15	1850
Ariyalur	Banana	Aug-15	1890
Ariyalur	Banana	Sep-15	1710
Ariyalur	Banana	Oct-15	1750
Ariyalur	Banana	Oct-15	1715
Ariyalur	Banana	Nov-15	1725
Ariyalur	Banana	Dec-15	1730

Figure A.2: Crop price estimation dataset

A.1.3 Crop yield estimation

The key features of this dataset as shown in fig A.3 are temperature, precipitation, soil organic matter (SOM), available water capacity (AWC), land area, vapor pressure deficit (VPD), and yield measured in tonnes per hectare. Temperature and precipitation are critical climatic factors that directly affect plant growth and development stages, while SOM and AWC are indicators of soil health and its ability to support crops through nutrient availability and water retention. VPD reflects the atmospheric conditions impacting plant transpiration and water usage efficiency, crucial for maintaining plant health and optimizing growth conditions.

Temparatu	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phosphorc	Fertilizer Na
20.87974	82.00274	38	Sandy	Maize	37	0	0	Urea
21.77046	80.31964	45	Loamy	Sugarcane	12	0	36	DAP
23.00446	82.32076	62	Black	Cotton	7	9	30	14-35-14
26.4911	80.15836	34	Red	Tobacco	22	0	20	28-28
20.13017	81.60487	46	Clayey	Paddy	35	0	0	Urea
23.05805	83.37012	35	Sandy	Barley	12	10	13	17-17-17
22.70884	82.63941	64	Red	Cotton	9	0	10	20-20
20.27774	82.89409	50	Loamy	Wheat	41	0	0	Urea
24.51588	83.53522	42	Sandy	Millets	21	0	18	28-28
23.22397	83.03323	33	Black	Oil seeds	9	7	30	14-35-14
26.52724	81.41754	28	Clayey	Pulses	13	0	40	DAP
23.97898	81.45062	48	Sandy	Maize	14	15	12	17-17-17
26.8008	80.88685	65	Loamy	Cotton	36	0	0	Urea
24.01498	82.05687	41	Clayey	Paddy	24	0	22	28-28
25.66585	80.66385	31	Red	Ground Nu	14	0	41	DAP
24.28209	80.30026	49	Black	Sugarcane	10	13	14	17-17-17

Figure A.3: Crop yield estimation dataset

A.1.4 Fertilizer recommendation

The dataset as shown in fig A.4 contain various environmental and agricultural factors, including temperature, humidity, soil moisture, soil type, and crop type. It also includes soil nutrient levels, specifically nitrogen, potassium, and phosphorous. The target variable is the fertilizer name, indicating the recommended fertilizer type. By analyzing these inputs, the model aims to optimize fertilizer recommendations, enhancing crop yields and maintaining soil health effectively. This approach helps farmers make informed decisions about fertilizer use, tailored to specific environmental conditions and crop needs.

Temp	Precipitati	SOM	AWC	Land Area	VPD	Yield(Tonnes/Hectare)
20.09499	58.196	1.246915	0.148338	436036.5	10.49023	30
20.08999	66.33406	1.464472	0.145533	424346	10.95782	30.9
20.46049	77.30545	1.477992	0.142567	623394.9	10.87587	49
19.56046	54.22876	1.386158	0.155162	571869.8	10.14754	55.2
20.23743	73.19876	1.34514	0.15229	583356	10.6434	53.2
19.55361	82.47753	1.20535	0.1445	866551.6	10.47122	34
20.24564	96.12793	1.008999	0.127869	383357.1	10.54908	17.3
20.33369	81.88613	1.346554	0.143753	622351.8	10.77626	26
19.88191	92.28505	1.178392	0.135271	410706.3	10.5428	29.7
20.57459	71.14455	1.152245	0.136125	466858.2	10.7608	40.3
20.45938	80.12195	1.371998	0.141116	507909.2	10.76189	42.2
19.87024	79.4635	1.346745	0.144469	461300.5	10.54793	41.2
20.322	88.55706	1.082417	0.133139	582474.9	10.70245	63.5
18.30538	97.88968	1.077956	0.147958	424314.4	9.966896	53.8
19.62504	106.0926	0.982532	0.130877	454170.5	10.54222	31.8

Figure A.4: Fertilizer recommendation dataset

A.2 Source code

A.2.1 Backend Code

Backend Code Snippet

import joblib

import requests, os

import pandas as pd

from flask import Flask, jsonify, request, render$_{t}emplate$

$fromflask_{c}orsimportCORS$

$fromsklearn.calibrationimportLabelEncoder$

$fromsklearn.discriminant_{a}nalysisimportStandardScaler$

$fromsklearn.imputeimportSimpleImputer$

$fromsklearn.model_{s}electionimporttrain_{t}est_{s}plit$

$frompredictorimportfertilizer_{p}rediction, yield_{p}rediction$

app = Flask('ML project')

cors = CORS(app)

@app.route('/', methods=['GET', 'POST'])

def index():

if request.method == 'POST':

Get form data

district = request.form['district']

n = float(request.form['n'])

p = float(request.form['p'])

k = float(request.form['k'])

print (n,p,k)

Get weather data from OpenWeatherMap API

weather$_{a}pi_{k}ey =' 4b69e9c09afabe33e9c0aa775a8400ce'$

$url = f'http://api.openweathermap.org/data/2.5/weather?q =$

$districtappid = weather_{a}pi_{k}ey'$

$response = requests.get(url).json()$

$print(response)$

$temp = response['main']['temp'] - 2734$

$humidity = response['main']['humidity']$

$rainfall = float(request.form['rainfall'])$

$print(temp, humidity)$

Load crop prediction model and predict crop

crop$_prediction_model = joblib.load('crop_predic.pkl')$

$crop_prediction_input = [[temp, humidity, rainfall, n, p, k]]$

$predicted_crop = crop_prediction_model.predict(crop_prediction_input)[0]print("PredictedCrop =$

$", predicted_crop)$

Load price estimation model and predict crop price

price$_estimation_model = joblib.load('trained_EST.pkl')$

$df = pd.read_csv('Price_Sorted_Modified.csv')$

$le_district = LabelEncoder()$

$df['District'] = le_district.fit_transform(df['District'])$

$le_crop = LabelEncoder()$

$df['Crop'] = le_crop.fit_transform(df['Crop'])$

$df['PriceDate'] = pd.to_datetime(df['PriceDate'], format =' df['Month'] = df['PriceDate'].dt.month$

$df['Year'] = df['PriceDate'].dt.year$

$df = df.drop('PriceDate', axis = 1)$

Split the dataset into training and testing sets

X$_train, X_test, y_train, y_test = train_test_split(df[['District','Crop','Month','Year']],$

$df['CropPrice(Rsperquintal)'], test_size = 0.2, random_state = 42)$

$imputer = SimpleImputer()$

$X_train = imputer.fit_transform(X_train)$

$X_test = imputer.transform(X_test)$

Feature scaling

scaler = StandardScaler()

X$_train = scaler.fit_transform(X_train)$

$X_test = scaler.transform(X_test)$

today = pd.Timestamp.today()
next$_six_months = pd.date_range(today, periods = 12,$
$freq =' MS').strftime("next_six_months_df = pd.DataFrame('District' : [le_district.transform([distr$

Use the trained XGBoost model to make predictions on the next 12 months dataset
next$_six_months_df = imputer.transform(next_six_months_df)$
$next_six_months_df = scaler.transform(next_six_months_df)$
$print("ALLGOOD!")$
$predicted_price = price_estimation_model.predict(next_six_months_df)$
$next12months = pd.date_range(today, periods = 12, freq =' MS')months = ['January',' February','$
$print(predicted_price)$
$price_data = ['month' : month.strftime('print(price_data)$
Send result to front end
result = it contains the information about the crops
print(result)
print(type(result.price$_data))$
$returnrender_template('result.html', result = result,$
$crop_info = crop_info)$
$else :$
$returnrender_template('index.html')$

@app.route('/predict/fertilizer', methods=['GET','POST'])
def prediction$_fertilizer() :$
$ifrequest.method ==' POST' :$
$GettheJSONdatafromthePOSTrequest$
$data = request.get_json()$

Extract features from the JSON data
temperature = data.get('temperature')
moisture = data.get('moisture')
humidity = data.get('humidity')
soil$_type = data.get('soilType')$

$crop_type = data.get('cropType')$
$N = data.get('N')$
$K = data.get('K')$
$P = data.get('P')$

predictions = fertilizer$_prediction(temperature, humidity, moisture, soil_type, crop_type, N, K, P)$
$ReturnthepredictionsasJSONresponse$
$print(predictions)$
$returnjsonify('prediction' : predictions)$
$else :$
$returnrender_template('predictions_fertilizer.html')$

@app.route('/predict/yield', methods=['GET','POST'])
def prediction$_yield() :$
$ifrequest.method ==' POST' :$
$GettheJSONdatafromthePOSTrequest$
$data = request.get_json()$

Assuming the JSON data contains keys corresponding to feature names
Extract features from the JSON data
Temp = data.get('Temp')
Precipitation = data.get('Precipitation')
SOM = data.get('SOM')
AWC = data.get('AWC')
Land$_Area = data.get('Land_Area')$
$VPD = data.get('VPD')$

Perform prediction using your model function
predictions = yield$_prediction(Temp, Precipitation, SOM, AWC, Land_Area, VPD)$

Return the predictions as JSON response
return jsonify('predictions': predictions)
else:
return render$_template('predictions_yield.html')$

A.2.2 Frontend code

Frontend code snippet

style
body
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
background-color: f4f4f4;

.container
max-width: 600px;
margin: 20px auto;
padding: 20px;
background-color: fff;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

h1
text-align: center;
margin-bottom: 20px;

label
display: block;
margin-bottom: 5px;

input[type="number"]
width: 100padding: 8px;
margin-bottom: 10px;
border: 1px solid ccc;
border-radius: 3px;

box-sizing: border-box;

button
width: 100padding: 10px;
background-color: 4CAF50;
color: fff;
border: none;
border-radius: 3px;
cursor: pointer;

button:hover
background-color: 45a049;

/style

script
function submitYieldForm()
const formData =
Temp: parseFloat(document.getElementById('Temp').value),
Precipitation: parseFloat(document.getElementById('Precipitation').value),
SOM: parseFloat(document.getElementById('SOM').value),
AWC: parseFloat(document.getElementById('AWC').value),
Land$_A$rea : parseFloat(document.getElementById('Land$_A$rea').value),
VPD : parseFloat(document.getElementById('VPD').value)
;
fetch('/predict/yield', method :' POST', headers : 'Content − Type' :' application/json', body : JSO
.then(response => response.json())
.then(data => //Handleresponsedataasneededconsole.log(data); alert('Yieldpredictionresult :' +JS
.catch(error => console.error('Error :', error); alert('Anerroroccurred, pleasetryagain.'););

< /script >
< /body >
< /html >