iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74365

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

AWS Cloud Enabled Crop Recommendation using Machine Learning Algorithms

Pooja G K[1], Anitha G[2]
[1]Student, Department of Computer science and Engineering, UBDTCE, Davanagere, Karnataka
[2]Associate Professor, Department of Computer science and Engineering, UBDTCE, Davanagere, Karnataka

Abstract: Agriculture plays a vital role in ensuring food security, yet farmers often face difficulties in selecting suitable crops due to varying soil and climatic conditions. This project introduces a cloud-enabled crop recommendation system that leverages machine learning techniques to provide reliable crop suggestions. The system processes key input parameters such as soil nutrients (Nitrogen, Phosphorus, Potassium, and pH) along with environmental factors (temperature, humidity, and rainfall) to determine the most suitable crops for cultivation. Two machine learning models, Random Forest and XGBoost, were developed and evaluated, with Random Forest demonstrating superior performance. To enhance accessibility and scalability, the best-performing model was deployed on Amazon Web Services (AWS), enabling farmers and stakeholders to access recommendations anytime and anywhere. By integrating machine learning with cloud computing, this work highlights the potential of data-driven solutions to support sustainable and efficient agricultural practices.

Keywords: Extreme Gradient Boosting (XGBoost), Random Forest Classifier (RFC), Machine learning (ML), Nitrogen (N), Phosphorus (p), Potassium (k), Amazon Web Services (AWS).

I. INTRODUCTION

Modern agriculture increasingly relies on data-driven decision-making to enhance productivity and sustainability. One of the key challenges farmer's face is selecting the most appropriate crop for cultivation under varying soil and climatic conditions. Agricultural productivity is heavily influenced by the selection of appropriate crops that match the soil and environmental conditions of a given region. Traditionally, farmers depend on experience, local practices, or generalized guidelines to make such decisions. Though, this strategy frequently results in suboptimal outcomes due to variations in soil fertility, nutrient levels, and climatic conditions. Inaccurate crop selection can result in reduced yield, inefficient resource utilization, and financial losses for farmers.The need for ecologically sustainable farming practices is growing along with the demand for food, there is an urgent need for systems that are intelligent can assist farmers in making data-driven decisions. Existing crop recommendation methods are often localized, lack scalability, and are not easily accessible to all farmers. Moreover, the absence of real-time decision support limits their practical usability.

Therefore, the problem addressed this project's deficiency of an accessible, accurate, and scalable crop recommendation system that includes rainfall, humidity, pH levels, and soil macronutrients (N, P, and K) into prediction models. By processing these parameters, the system determines which crop is best suited for a certain environment. Deploying the model on a cloud platform ensures scalability, real-time accessibility, and seamless integration for end-users. Such a system not only supports precision farming but also empowers farmers with scientific insights, thereby improving yield outcomes and resource efficiency. A cloud-enabled system is needed to bridge this gap, providing farmers with reliable suggestions that can improve output, optimize resource usage, and contribute to sustainable agriculture.

This work covers the entire pipeline, which includes preparing the data, feature extraction, model training, validation, and deployment on a cloud platform. Cloud integration ensures that the solution remains scalable, platform-independent, and easily accessible to end-users in real time. Because of its adaptability to various agricultural regions, the system can be expanded with further features like temperature, soil moisture, or micronutrient levels in future enhancements.

However, this research is limited to the available dataset and predefined crop categories used during training. External factors such as market demand, pest outbreaks, government policies, and sudden climatic anomalies are not considered within the present scope. Despite these limitations, the system provides a strong foundation for data-driven crop recommendation and serves as a step toward intelligent, technology-driven farming practices. From a research standpoint, this work demonstrates the effective application of data science in agriculture, helping to develop technologies for smart farming. It also establishes a foundation for additional study, where additional parameters such as temperature, soil moisture, and pest patterns can be incorporated.

Overall, this research is crucial for more than just farmers and the agricultural community but also for policymakers, researchers, and society at large by promoting sustainable food security and technological innovation in farming.

A. Objectives of this work

The main objectives of this work are as follows:

1) To create a cloud-enabled crop recommendation system that identifies the best crop depending on climate and soil conditions such as Nitrogen (N), Phosphorus (P), Potassium (K), pH, rainfall, and humidity.
2) To apply ML methods for processing agricultural data, feature extraction, and classification in order to attain precise and trustworthy crop predictions.
3) To deploy the trained model on a cloud platform, ensuring scalability, accessibility, and real-time interaction for farmers and stakeholders.
4) To promote precision farming practices by minimizing guesswork in crop selection and enhancing resource efficiency.
5) To improve agricultural productivity and sustainability by providing data-driven recommendations that support better decision-making in diverse environmental conditions.

II. LITERATURE REVIEW

Dalavai et al. (2024) developed a method that utilizes multiple ML algorithms to support farmers in selecting suitable crops. The study carried out a comparative analysis of five algorithms—Logistic Regression, One-vs-All, Decision Trees, Bagging Classifier, and Random Forest—to evaluate their effectiveness in predicting crops under varying agricultural conditions. The authors emphasized how crop recommendation is essential in overcoming real-world challenges such as unpredictable weather, soil variability, and changing market demands. Their outcomes depict that ML may greatly increase crop prediction efficiency and accuracy, thereby helping farmers maximize yield and profitability. Moreover, the web-based nature of their system makes it more accessible and practical for stakeholders, providing insightful information on the potential applications of various algorithms for reliable, real-time agricultural decision-making [1].

A hybrid machine learning method for forecasting Sri Lankan crop output is presented by Munasinghe, H. N. et al., (2024). Eleven districts and main crops were included in the study, along with meteorological variables including soil moisture and humidity. The KNN–Random Forest combo produced the best accuracy out of all the models that were tested. Additionally, a basic Flask web application was created to give farmers real-time yield forecasts [2].

Muddarla, B and Vatti, P. R, (2024) focused on the integration of SQL and Python in cloud-based machine learning. Python enables flexible transformation, model creation, and visualization [3].

D. Bayazitov et al., (2024) proposed an AWS-based cloud storage solution integrated with AI. It demonstrates how AWS provides scalability, dependability, and cost benefits. AI aids in data analysis, outcome visualization, and insight discovery. It also examines solutions to problems like security and increasing data demands [4].

Kandi, A and Basani, M, (2024), examined the efficiency of AWS and Google Cloud in handling real-time analytics and machine learning workloads. Using high-velocity transactional data, the research evaluated both platforms across several performance dimensions, including data ingestion latency, query processing, inference speed, scalability, and cost-effectiveness. Findings revealed that AWS consistently delivered better results. It recorded a lower ingestion latency (116.1 ms) compared to Google Cloud (125.2 ms) and demonstrated a notable 10% improvement in query execution times. In terms of machine learning integration, AWS SageMaker achieved faster inference (75.5 ms) than Google's AI Platform (89.6 ms), which is vital for applications that need real-time forecasts, such as fraud detection and recommendation engines. The study also highlighted AWS's superior scalability, sustaining reliable performance beyond 2,500 records per second, while Google Cloud began to plateau around 2,000 records per second. These results underline AWS as a stronger and scalable choice for data-intensive, real-time environments where high throughput and low latency are critical [5].

Barvin P and Sampradeepraj, (2023), investigated crop recommendation utilizing two graph-based deep learning models, GCN and GNN. The algorithm used soil parameters to recommend appropriate crops. The findings demonstrated that while GNN handled more general interactions, GCN was superior at collecting local feature correlations. The study emphasizes how graph-based models can enhance resource efficiency and precision farming [6].

Rachid Ed-daoudi et al., (2023), proposed a machine learning web-based crop recommendation system for Morocco. It draws attention to how AI and precision farming can increase yields while lowering manual labor costs. For more precise forecasts, the study recommends improving the platform with IoT and real-time monitoring [7].

Hasan et al., (2023), developed an ensemble machine learning–based crop recommendation system utilizing agricultural and meteorological datasets from Bangladesh. After comparing a number of ensemble and classical algorithms, they developed a novel ensemble technique called KRR that performed better than the competition in terms of accuracy and error reduction. According to their findings, wheat production was declining while rice and potato production was on the rise. A recommender system to identify the good crops to grow in future seasons was also suggested by the study [8].

Siddhant Ashok Doke, (2023), presented an overview of AWS as a leading platform in cloud computing. The study emphasizes AWS's role in offering on-demand services such as compute power, storage, and databases, which enable businesses to scale resources dynamically. A significant focus is placed on Amazon EC2 and AWS Lambda. EC2 allows scalable virtual servers with various setups for diverse workloads, while Lambda supports serverless computing, enabling consumers to execute apps without having to worry about server management [9].

Tillakaratne, N. N et al., (2022) focuses on cloud-enabled crop recommendation platform designed to support precision farming through the incorporating machine learning. The study emphasizes the growing Artificial intelligence's role in agriculture, particularly in enhancing farmers' ability to make timely and accurate decisions. To establish its significance, the authors provide an outline of precision farming practices and compare their work with that of other recent studies in the field. Their proposed platform highlights how cloud-based technologies can serve as an accessible and cost-free solution for farmers, while also encouraging researchers to innovate further in this domain. By combining machine learning with cloud infrastructure, the work demonstrates a scalable approach for delivering intelligent agricultural tools that can transforming conventional farming into a more data-driven and efficient process [10].

S. Abarna and Priya Ganesh (2022) developed deep learning-based models to evaluate how the underlying algorithms perform with respect to different performance criteria. The algorithms evaluated in this study are the XGBoost machine learning (ML) algorithm, Convolutional Neural Networks (CNN), XGBoost, and Recurrent Neural Networks (RNN). For the case study, they predicted crop yield based on the environmental, soil, silt, nitrogen, clay, ocd, ocs, pHH2O, sand, soc, ceo, water and crop parameters has been a potential research topic. The proposed method has high performance by preforming feature selection on predicting the crop yield. The proposed method used for both the soyabeans and corn crop for predicting their yield [11].

Gosai, D et al., (2021) developed a crop recommendation system aimed at maximizing crop yield by leveraging machine learning techniques. Their work primarily utilized Support Vector Machines (SVM) and Artificial Neural Networks (ANN) to build predictive models. The authors trained their structure utilizing soil sampling data collected from laboratory analyses, allowing the models to recommend the ideal crops for particular farming locations according to the type of soil. This approach emphasized the significance of soil characteristics as key determinants in agricultural decision-making. The study showed how ML can effectively bridge soil science and crop planning, ultimately helping farmers to make wise decisions to increase productivity. The outcomes taken from this work are particularly relevant to our project, as it reinforces the value of data-driven crop recommendation systems, while also showcasing the applications of different machine learning methods in similar agricultural contexts [12].

Taranjot Singh (2021) examined the transformative role of Amazon Web Services (AWS) in the field of cloud computing. The study highlights how AWS offers a broad variety of services, including as databases, storage, and computation, networking, security, analytics, and automated scaling, which collectively empower organizations to optimize their IT operations. Businesses can use these services to minimize infrastructure costs, increase scalability, and improve operational efficiency. Singh also explains the AWS Cloud Adoption Framework (CAF), which provides structured guidance and optimal methods for identifying organizational gaps and streamlining processes. This framework supports businesses in adopting cloud technologies more effectively across their IT lifecycle. With its reliability and flexibility, AWS has become a trusted platform not only for large enterprises but also for start-ups, particularly in areas such as data storage, analytics, and archiving. The study underlines AWS's pivotal role in driving modern cloud adoption and shaping the digital strategies of diverse organizations [13].

According to Rajshree Swarnkar et al., (2021), AWS provides the ability to automate manual security tasks, allowing you to focus on growing and automating your organization [14].

Sonal Agarwal and sandhya Tarar (2021), proposed a model which enhanced by applying deep learning techniques and along with the prediction of crop, a clear information is achieved regarding the amounts of soil ingredients needed with their expenses separately. It provides a better accuracy than the existing model. It analyzes the given data and help the farmers in predicting a crop which in return help in gaining profits. The climatic and soil conditions of land are taken into consideration to predict a proper yield. The objective is to present a python based system that uses strategies smartly to anticipate the most productive reap in given conditions with less expenses. In this paper, SVM is executed as Machine Learning algorithm while LSTM and RNN are used as Deep Learning algorithms [15].

III. PROPOSED WORK

The methodology describes the step-by-step approach used to develop the proposed cloud-enabled crop recommendation system. It includes data collection, pre-processing, feature selection, ML model training, and deploying it in the cloud. The process aims to ensure precision, scalability, and immediate access for farmers and stakeholders.

A. *System Workflow*

The system workflow consists of five major phases:

1) Data Collection: Agricultural datasets with soil macronutrients (N, P and K), pH levels, rainfall, and humidity are gathered from publicly available repositories and agricultural research centers.
2) Data Pre-processing: The dataset is cleaned, normalized, and encoded to ensure consistency. This step removes missing or noisy values.
3) Feature Engineering and Selection: Parameters like N, P and K, pH, rainfall, and humidity are identified as key features that influence crop yield.
4) Model Training and Validation: XGBoost, Random Forest are the ML models which are trained and evaluated. Calculated the performance metrics includes F1-score, recall, accuracy, and precision. Model with best performance is deployed to cloud.
5) Cloud Deployment: The model that performs the best is used on a cloud platform like AWS. It has a web-based interface for real-time crop recommendations.

The workflow diagram of methodology is shown in below Fig 1.

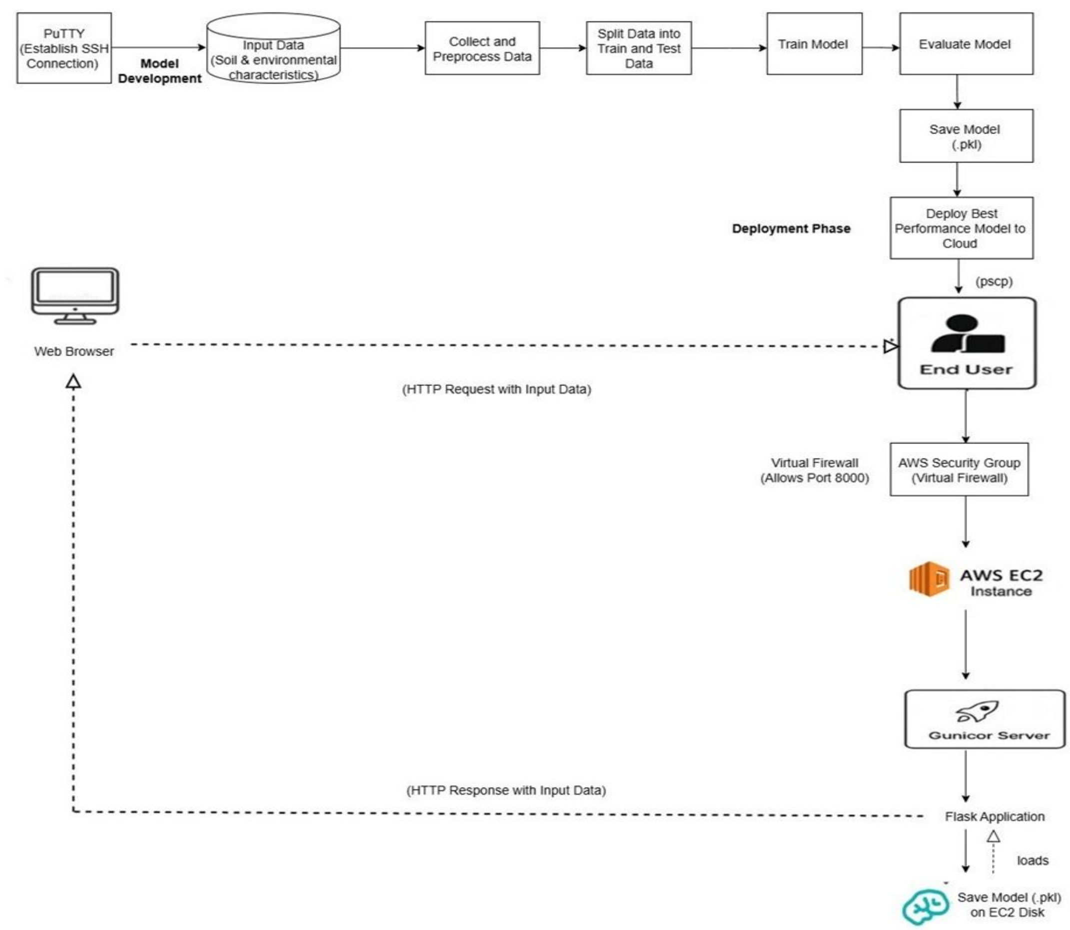


Fig 1: Workflow diagram of methodology

The proposed system for crop recommendation is developed and deployed in three main stages: Model Development, Deployment, and User Interaction. Each stage plays a specific role in building, hosting, and delivering the application to the end users.

1) Model Development

The first stage of the project takes place on the developer's personal computer. Here, the required Python code is written, which includes the crop recommendation logic, the Flask web application (app.py), and supporting files such as index.html and style.css for the user interface. At the same time, a machine learning model is trained utilizing soil and environmental characteristics as input data. Sets for training and testing are produced from the pre-processed dataset and the trained model is saved as a .pkl file. To manage dependencies properly, the entire project is organized within a virtual environment (myprojectenv) that contains the Flask code, templates, static files, and the trained model.

2) Deployment

Once the application and model are ready, they are moved to the cloud for deployment. An AWS EC2 instance is set up, which acts as a virtual server running Ubuntu. A secure SSH connection is established using PuTTY and private key authentication. Next, the project files are copied from the local system to the cloud instance using pscp.

To make the application accessible online, the AWS Security Group is adjusted to allow incoming traffic on port 8000, serving as a virtual firewall. On the EC2 server, the Gunicorn server runs and serves the Flask application. The saved machine learning model (.pkl file) is stored on the EC2 disk, ready to be loaded whenever a prediction request comes in.

3) User Interaction

In the final stage, the deployed application becomes available to users. An end user can open a web browser and access the application by entering the public IP address of the EC2 instance along with port 8000. The request first goes through the configured Security Group, which checks and forwards it to the EC2 instance. The Gunicorn server processes this request and sends it to the Flask application.

The Flask app loads the crop recommendation model, processes the input values (soil and environmental parameters), and generates predictions about the most suitable crop. The response, along with the designed HTML interface, is then sent back to the user's browser, completing the interaction. The workflow above ensures a smooth process, starting from developing the model locally, to deploying it securely on AWS, and finally delivering real-time crop recommendations to farmers or end users through a simple web interface.

B. Dataset

This section describes the dataset used in this work.

1) This project's dataset was gathered from Kaggle.
2) Features used: Nitrogen (N), Phosphorus (P), Potassium (K), pH, Temperature, Humidity, Rainfall.
3) Size of dataset: There are 2,200 records with 8 attributes in the dataset utilized for this research. The input attributes include soil nutrients (Nitrogen, Phosphorus, Potassium), environmental factors (temperature, humidity, rainfall), and soil pH value. The target attribute is the crop label, which specifies the most suitable crop for the given conditions. This combination of soil and climatic features provides a complete dataset for evaluating and training the crop recommendation model.

C. Methods

This session describes methods used in the project:

1) Data collection, data cleaning, normalization, and handling missing values.
2) Dataset splitting: The dataset was divided into two sets: a training set and a testing set in order to assess the crop recommendation model's performance. A train-test split ratio of 80:20 was used, where 80 percent of the data was for training the model to capture patterns and relationships among the features. The remaining 20 percent was set aside for testing. This separation ensures that the model is evaluated on new data, providing a more reliable measure of how well it can generalize.
3) Model Training: The model was trained using the Random Forest and XGBoost algorithms.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

D. Algorithms

The algorithms used in this work are XGBoost and RFC.

1) XGBOOST Algorithm:

Among the most popular machine learning techniques for classification and regression tasks is called Extreme Gradient Boosting, or XGBoost. It improves on the basic gradient boosting framework to provide better accuracy and efficiency compared to older methods. XGBoost is widely used in academic research and industry because it is fast, scalable, and capable of handling large and complex datasets.

At its core, XGBoost generates a group of decision trees in a step-by-step manner. Each new tree is built to fix the errors created by the earlier trees which improves overall model performance. The approach uses gradient descent to optimize a loss function, such as log loss for classification or mean squared error for regression. It also adds regularization to avoid overfitting.

One of the main strengths of XGBoost is its regularization techniques (L1 and L2). These techniques control model complexity and make the algorithm more robust. Additionally, XGBoost supports parallel processing, handles missing values, and includes tree pruning, making it faster and more efficient than older gradient boosting methods.

The working of XGBoost is given in the phases listed below:

a) Step 1: Initialization- The process begins with an initial prediction, usually the average value of the target variable for regression or equal probability for classification.
b) Step2: Error Calculation- The algorithm computes the difference (residuals) between the predicted and actual values.
c) Step 3: Gradient and Hessian Computation- XGBoost calculates the gradient (first derivative) and hessian (second derivative) of the loss function. These calculations guide the direction and strength of necessary improvements.
d) Step 4: Tree Construction- A new decision tree is built to predict the residual errors, with splits chosen to maximize information gain.
e) Step 5: Model Update- The predictions from the new tree are added to the existing model using a learning rate (shrinkage) to control the step size.
f) Step 6: Iteration- Steps 2 to 5 are repeated for a set number of rounds or until the loss function stops improving.
g) Step 7: Final Prediction- The ensemble of all decision trees provides the final prediction.

Overall, XGBoost achieves high predictive accuracy by combining boosting, regularization, and efficient computation. Its capacity to manage missing data, imbalanced datasets, and large-scale problems makes it among the most powerful algorithms in modern machine learning.

2) Random Forest Algorithm and its working:

One well-known machine learning method that is a part of the supervised learning approach is Random Forest. In machine learning, it can be applied to both classification and regression issues. Its foundation is the idea of ensemble learning, which is the process of merging several classifiers to solve a challenging issue and enhance the model's performance. As the name implies, “Random Forest is a classifier that uses a number of decision trees on different subsets of the dataset and averages them to increase the dataset's predicted accuracy”. Rather than depending on a single decision tree, the random forest forecasts the final result by taking the predictions from each tree and calculating the majority vote of predictions. Accuracy is improved and overfitting is prevented because the forest has more trees. The random forest is created by combining N decision trees in the first phase, and predictions are generated for each tree created in the second phase.

The steps listed below can be utilized to explain the working process:

a) Step-1: Choose K data points at random from the training set.
b) Step-2: Construct the decision trees linked to the chosen data points.
c) Step-3: For the decision trees you wish to construct, select the number N.
d) Step-4: Repeat Step 1 & 2.
e) Step-5: Determine each decision tree's predictions for the new data points, then allocate them to the category with the most votes.

Using random data and features for every tree helps avoid overfitting and makes the overall prediction more accurate and trustworthy.

E. Cloud Resources

Cloud resources used in this work are AWS EC2 and Security Groups.

1) Amazon EC2 Instance

Amazon Elastic Compute Cloud (EC2) is a cloud-based virtual server that provides scalable computing resources. It enables users to run applications on a flexible infrastructure without the need for physical hardware. In this work, the EC2 instance serves as the deployment environment for hosting the Flask application and the trained machine learning model.

2) AWS Security Group

An AWS Security Group acts as a virtual firewall that controls inbound and outbound traffic to EC2 instances. It defines a set of rules that determine which type of network traffic is allowed to reach the server. In this project, the security group is configured to allow HTTP requests on port 8000 in order for end users to have access to the deployed web application.

IV. IMPLEMENTATION

The implementation of the crop recommendation application involves three main phases: Model Development, Deployment on AWS, and User Interaction.

A. Model Development

This is where everything begins on your personal computer. In this phase, you design and build the application step by step:

1) *Input Data Collection:* The process starts by gathering data related to soil characteristics (such as nitrogen, phosphorus, potassium, and pH) and environmental factors (including temperature, humidity, and rainfall). These inputs form the foundation for predicting suitable crops.
2) Data Pre-processing and Splitting: The collected data is cleaned and pre-processed to handle missing values and ensure consistency. It is then divided into training and testing sets, enabling the model to learn patterns while also being evaluated on unseen data.
3) Model Training and Evaluation: Many ML models comprise Random Forest, SVM, decision trees are trained on the processed data. Their performance is tested using evaluation metrics to determine the best-fit model for making accurate crop recommendations.
4) Model Saving: The finalized model, along with pre-processing tools, is saved as files. These are later used in deployment so that the application can make predictions without retraining the model each time. The trained models and pre-processors (like StandardScaler, LabelEncoder) are then saved as .pkl files for reuse.

- Write the Python Flask code along with HTML templates and CSS files to create the web interface.
- All files are neatly arranged into a project directory with separate folders for templates, static files, and models.
- Machine Learning Model Training: this step involves training the machine learning models (e.g., Decision Tree, Random Forest) using historical data and save them along with data pre-processors (e.g., StandardScaler, LabelEncoder) as .pkl files.
- A Python virtual environment (venv) is created to manage the project dependencies, ensuring the code runs consistently without conflicts.

B. Deployment on AWS Cloud

Once the application is ready, the next step is to move it to the cloud so anyone can access it.

Amazon Web Services (AWS) is the world’s leading cloud computing platform developed by Amazon. It allows individuals, startups, and large organizations to access computing resources on demand. Instead of buying expensive servers or hardware, users can rent them virtually through AWS. The platform offers an extensive array of services, including storage, databases, and networking, analytics, and AI tools. Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) are among the services it offers. This implies that anything from basic websites to extensive enterprise systems can be powered by AWS. One of the biggest advantages of AWS is scalability. One can increase or decrease resources depending on your needs without investing in new hardware. It is therefore perfect for companies with varying workloads.

AWS is also called for its cost-effectiveness. You only pay for what you use because it operates on a pay-as-you-go basis. This eliminates the upfront investment required for traditional IT infrastructure. In terms of security, AWS offers strong encryption, compliance certifications, and data protection. It is trusted by industries like banking, healthcare, and government agencies.

AWS Data centers are dispersed throughout the globe, ensuring high availability and reliability. Some popular AWS services include EC2 (servers), S3 (storage), RDS (databases), and Lambda (serverless computing). AWS also supports new techniques like machine learning, Internet of Things (IoT), and cloud-native development. Overall, AWS empowers organizations to innovate faster, reduce costs, and stay competitive. It has become the backbone of many digital businesses worldwide.

C. User Interaction Phase

This phase describes how an end-user interacts with your deployed application.

1) *User Initiates Request:* A user opens a web browser and enters the EC2 instance's Public IPv4 address followed by the configured port. http://<EC2-Public-IP>:8000
2) *Traffic Flow via Security Group:* The incoming HTTP request travels across the internet and first reaches the AWS Security Group. The Security Group's rules are evaluated, and since port 8000 is open, the request is permitted to reach the EC2 instance.
3) *Gunicorn Receives Request:* The Gunicorn server, running on the EC2 instance, receives the incoming HTTP request.
4) Request Hand-off to Flask Application: Gunicorn forwards the request to your Flask application (app.py), which processes the specific route requested (e.g., the home page /).
5) *ML Model Inference:* The Flask application loads the necessary .pkl files (models and pre-processors) from the instance's storage. It then processes user input (if any) and uses the loaded machine learning model to generate a prediction or recommendation.
6) *HTML Template Rendering:* Flask uses Jinja2 to render the index.html template. During this process, Flask's url for function dynamically generates correct paths for static assets (like style.css and crop.jpg) located in the static/ directory.
7) *Response Sent to User's Browser:* The rendered HTML, along with separate requests for CSS and image files (which are also served by Gunicorn from the static directory), is sent back to the user's web browser.
8) *Webpage Display:* The user's browser combines the HTML, CSS, and images to display the fully functional and styled crop recommendation web application.

V. RESULTS AND DISCUSSIONS

The experimental results are shown below:

1) *Cloud Enabled Crop Recommendation Register page.*

The registration page of Crop recommendation WebApp is shown below in Fig 2.

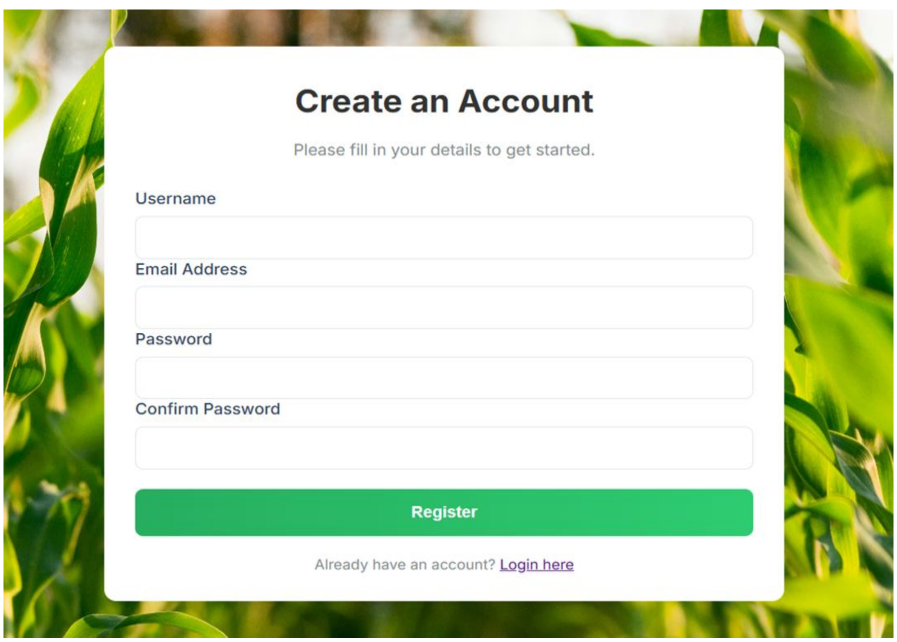


Fig 2: Register page of Crop Recommendation WebApp.

2) Cloud Enabled Crop Recommendation login page.

The login page of Cloud Enabled Crop Recommendation WebApp is shown below in Fig 3.

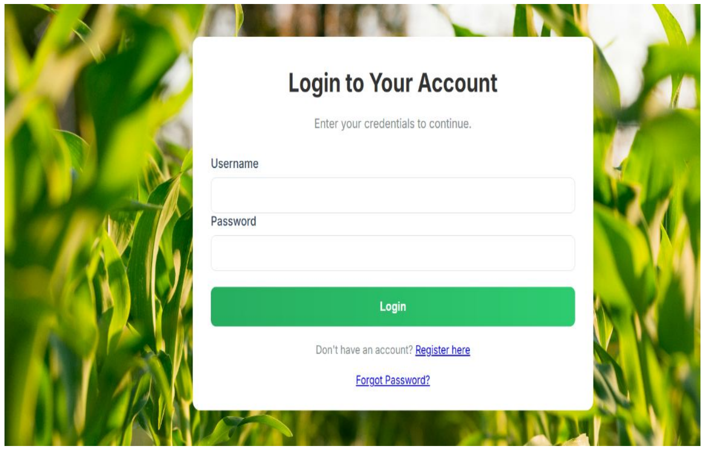


Fig 3: Login page of Crop Recommendation WebApp.

3) Crop Prediction with confidence Percentage.

Crop Prediction Result for Coffee with confidence is shown in Fig 4.

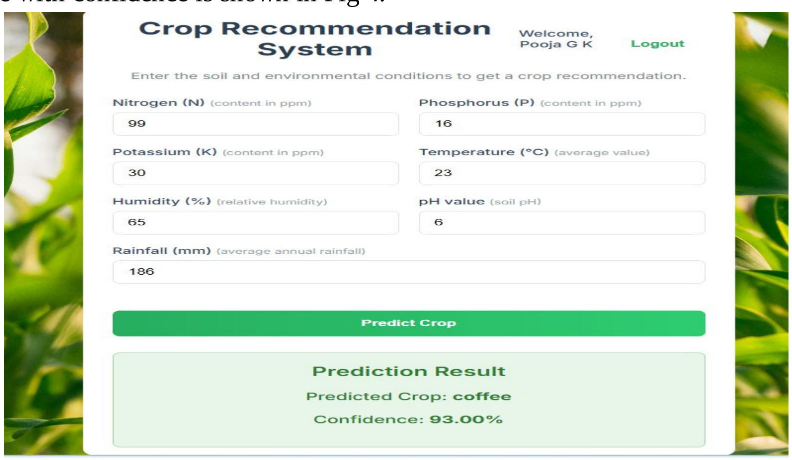


Fig 4: Crop Prediction Result for Coffee with confidence 93%.

4) Crop Prediction User Interface.

Crop Prediction Result for lentil with confidence is shown in Fig 5.

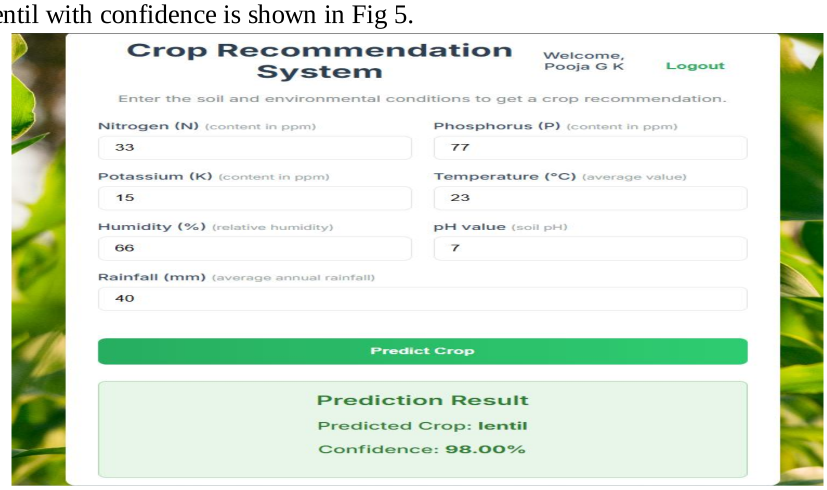


Fig 5: Crop Prediction Result for lentil with confidence 98%.

Two well-known machine learning models were utilized to assess the crop recommendation system: Extreme Gradient Boosting (XGBoost) and Random Forest Classifier (RFC). Both models were trained and tested on the prepared dataset, and Standard evaluation parameters, such as accuracy, precision, recall, and F1-score, were used to gauge their performance.

A. *XGBoost Results*

The XGBoost model showed a high degree of predictive ability with a 98.64% accuracy rate. It had a precision of 98.69%, which means most of the crops it recommended were relevant to the given conditions. The recall of 98.64% indicated that the model correctly identified the majority of suitable crops. The F1-score of 98.63% reflects the balanced the model's performance in terms of both precision and recall. These results highlight how effective XGBoost is at understanding complex relationships among soil and climate attributes. The Fig 6 shows Performance metrics of XGBoost algorithm.

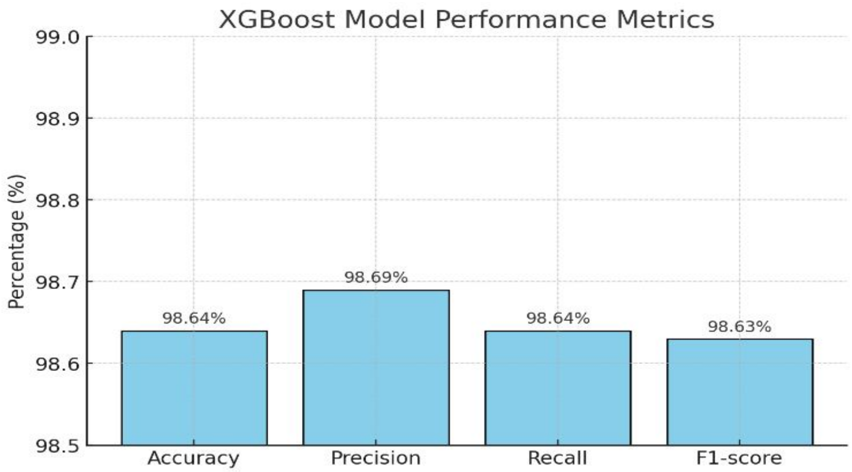


Fig 6: Performance metrics of XGBoost algorithm

B. *Random Forest Classifier Results*

The Random Forest Classifier (RFC) slightly outperformed XGBoost in all evaluation metrics. RFC attained the maximum level of accuracy of 99.32%, suggesting that nearly all predictions were correct. With a precision of 99.37%, the model demonstrated reliability in recommending the right crops. The recall of 99.32% indicated that the model successfully identified most appropriate crops without significant misclassifications. The F1-score of 99.32% confirmed that RFC maintained a strong balance between precision and recall, reinforcing its strength as a classification model. The fig 7 shows Performance metrics of Random Forest Classifier algorithm.

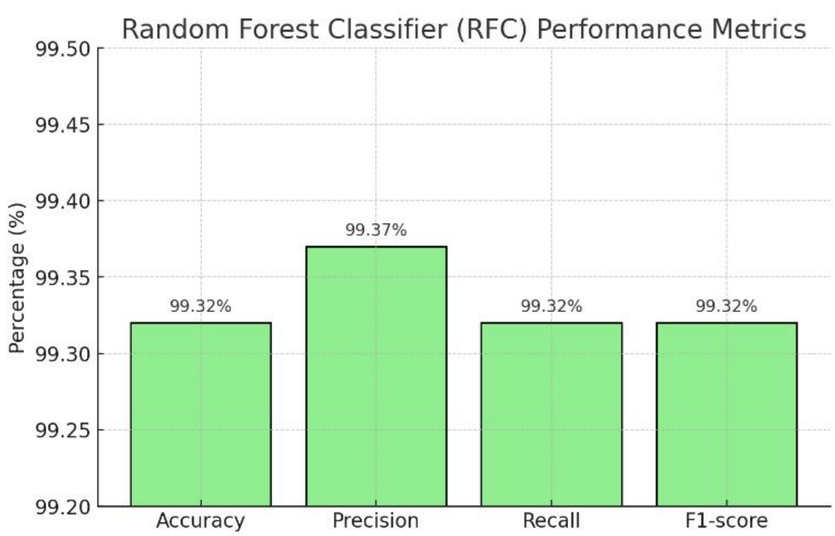


Fig 7: Performance metrics of Random Forest Classifier algorithm

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

C. Comparative Analysis

When the two models are compared, they both produced results with good precision and dependability with performance values above 98%. However, in every metric, the RFC performed marginally better than XGBoost. This minor variation implies that RFC is better suited for this dataset, as it captures feature interactions more effectively and reduces prediction variance through ensemble averaging. The Fig 8 shows Performance comparison of XGBoost and Random Forest Classifier algorithms.

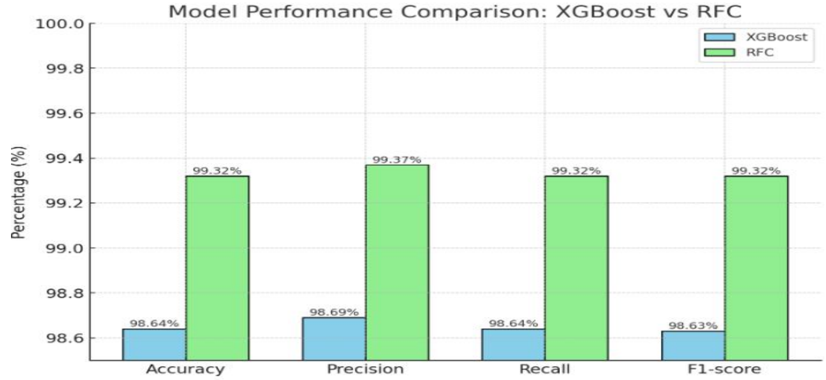


Fig 8: Performance comparison of XGBoost and Random Forest Classifier algorithms

D. Deployment Decision

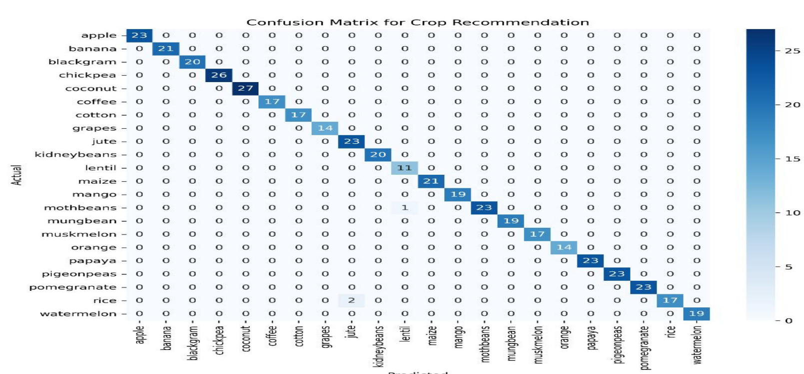
The RFC was selected as the last model to be utilized in a cloud-based setting due to its exceptional performance. This allows end-users to access the crop recommendation system in real-time, benefiting from the high accuracy and stability of the model. The cloud deployment also makes the system scalable, enabling farmers and stakeholders to use it for decision-making across various regions and datasets.

E. Confusion Matrix

The confusion matrix of the Random Forest Classifier (RFC) reflects a high level of predictive accuracy across multiple crop categories. The majority of predictions lie on the diagonal, indicating correct classification, with very few misclassifications observed. For example, crops such as apple, banana, coconut, pigeonpeas, and pomegranate achieved perfect classification with no false predictions. There were a few minor mistakes for crops like mothbeans (1 misclassified sample) and rice (2 misclassified samples), however, in comparison to the total number of predictions, these variances are insignificant.

From the matrix, the overall accuracy of the model exceeds 99%, indicating that almost all forecasts match the true crop labels. Furthermore, the precision for most classes is close to 1.0, as false positives are nearly absent. Similarly, recall is very high since the majority of actual crop instances are correctly identified, with only a few crops experiencing minimal under-prediction. The F1-score, which balances both precision and recall, stays constant at a high level in every crop category, reinforcing the reliability of the classifier.

These metrics collectively validate the effectiveness of the RFC model in handling complex, multi-class agricultural datasets. The strong classification performance ensures that the system can provide farmers with accurate crop recommendations, ultimately supporting informed agricultural decision-making. The Fig 9 shows Confusion Matrix for Random Forest Algorithm.



`Fig 9: Confusion Matrix for Random Forest Algorithm.

VI. CONCLUSION

This study focused on creating and implementing a smart crop recommendation system based on ML, specifically the Random Forest Classifier (RFC) algorithm, and deploying it on the AWS cloud platform. The results showed that the RFC model achieved excellent predictive performance, with a 99.32% total accuracy rate, as well as good F1-score, precision, and recall scores. These findings indicate how well RFC handles agricultural datasets where factors like soil nutrients (Nitrogen, Phosphorus and Potassium), pH, temperature, humidity, and rainfall influence crop suitability.

A key contribution of this work is the successful cloud deployment on AWS, which ensures the model is accurate and also scalable, accessible, and useful for real-world applications. Compared to local machine learning experiments, deploying the system on AWS offers several benefits: it can handle large datasets, provides secure access from remote locations, and reliably delivers real-time recommendations. This makes the system especially valuable for farmers, policymakers, and agricultural extension workers who need timely and accurate insights for making informed decisions.

Furthermore, merging machine learning and cloud computing takes another step towards digital transformation in agriculture. By using cloud services, the proposed system makes advanced technologies available even in rural or resource-limited areas, thereby connecting AI research with its practical benefits for farming.

In summary, the project shows how machine learning-based crop recommendations, backed by cloud infrastructure, can improve agricultural productivity, optimize resource use, and support sustainable farming practices. The RFC model's excellent performance and also the reliability of AWS deployment highlights how well this method works to address real-world agricultural challenges.

VII. FUTURE SCOPE

The current crop recommendation system is built on a static dataset and offers a strong starting point. However, it can be expanded into a more advanced and practical solution in the future.

REFERENCES

[1] Dalavai, Adarsha & Dalli, Manvith & Jogi K, Dhanush & H M, Monisha. (2024). A Web Based Crop Recommendation System Using Various Machine Learning Algorithms.

[2] H. N. Munasinghe, E. G. T. Dasunika, and W.W.L.Subhodani, "A Hybrid Approach for Crop Yield Prediction using Machine Learning Algorithms," Int. J. Soc. Stat., vol. 1, no. 02, 2024, doi: 10.31357/ijss.v1i02.8274.

[3] B. Muddarla and P. R. Vatti, "Machine Learning in Cloud Environments : Leveraging SQL and Python for Big Data Analytics," vol. 7, no. 7, pp. 12–21, 2024.

[4] D. Bayazitov, K. Kozhakhmet, A. Omirali, and R. Zhumaliyeva, "Leveraging Amazon Web Services for Cloud Storage and AI Algorithm Integration: A Comprehensive Analysis," Appl. Math. Inf. Sci., vol. 18, no. 6, pp. 1235–1246, 2024, doi: 10.18576/amis/180606.

[5] A. Kandi and M. Anurag Reddy Basani, "Real-time Analytics on AWS and Google Cloud to Unlock Data Driven Insights," Int. J. Sci. Res., vol. 13, no. 11, pp. 302–308, 2024, doi: 10.21275/sr241105211600.

[6] P. Ayesha Barvin and T. Sampradeepraj, "Crop Recommendation Systems Based on Soil and Environmental Factors Using Graph Convolution Neural Network: A Systematic Literature Review †," Eng. Proc., vol. 58, no. 1, 2023, doi: 10.3390/ecsa-10-16010.

[7] R. Ed-daoudi, A. Alaoui, B. Ettaki, and J. Zerouaoui, "A Predictive Approach to Improving Agricultural Productivity in Morocco through Crop Recommendations," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 3, pp. 199–205, 2023, doi: 10.14569/IJACSA.2023.0140322.

[8] M. Hasan et al., "Ensemble machine learning-based recommendation system for effective prediction of suitable agricultural crop cultivation," Front. Plant Sci., vol. 14, no. August, pp. 1–18, 2023, doi: 10.3389/fpls.2023.1234555.

[9] S. A. Doke, "A Review on AWS - Cloud Computing Technology," Int. Res. J. Mod. Eng. Technol. Sci., no. 06, pp. 3000–3005, 2023, doi: 10.56726/irjmets42351.

[10] N. N. Thilakarathne, M. S. A. Bakar, P. E. Abas, and H. Yassin, "A Cloud Enabled Crop Recommendation Platform for Machine Learning-Driven Precision Farming," Sensors, vol. 22, no. 16, 2022, doi: 10.3390/s22166299.

[11] S.ABARNA, & PRIYA, P.GANESH. (2022). CROP YIELD PREDICTION USING DEEP XGBOOST ALGORITHM. International journal of engineering technology and management sciences. 297-304. 10.46647/ijetms.2022.v06i05.043.

[12] D. Gosai, C. Raval, R. Nayak, H. Jayswal, and A. Patel, "Crop Recommendation System using Machine Learning," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 558–569, Jun. 2021, doi: 10.32628/CSEIT2173129.

[13] T. Singh, "The effect of Amazon Web Services (AWS) onCloud-Computing," Int. J. Eng. Res. Technol., vol. 10, no. 11, pp. 480–482, 2021.

[14] R. Swarnkar, S. Jain, and M. Kusum, "AWS Security Issues And Good Practices," An Int. Peer Rev. Journal), www.ijaconline.com, vol. XV, no. June, pp. 1–8, 2021, [Online]. Available: www.ijaconline.com.

[15] Agarwal, Sonal & Tarar, Sandhya. (2021). A HYBRID APPROACH FOR CROP YIELD PREDICTION USING MACHINE LEARNING AND DEEP LEARNING ALGORITHMS. Journal of Physics: Conference Series. 1714. 012012. 10.1088/1742-6596/1714/1/012012.