Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 46 No. 1 (2025)

From Soil Health to Harvest: Comprehensive Approaches to Crop Productivity Enhancement

[1]N. Niharika, [2]Dr. N. Swapna, [3]Dr. N. Arjun, [4]Dr. Raju Dara

[1]Research Scholar, Department of Computer Science, Vignana Bharathi Institute Of Technology, Hyderabad,

[2]Associate Professor, Department of Computer Science, Vignana Bharathi Institute Of Technology, Hyderabad,

[3]Associate Professor, Department of Computer Science, Vignana Bharathi Institute Of Technology, Hyderabad,

[4]Professor, Department of Computer Science, Vignana Bharathi Institute Of Technology, Hyderabad,

Abstract:

In today's world, technology is essential in various sectors for overcoming challenges and achieving optimal results. In India, farming holds a vital position in shaping the nation's economic foundation, engaging almost half of the workforce.. However, this sector is largely dependent on natural conditions and faces numerous practical challenges. Traditional farming methods are still prevalent, and technological advancements have been slow to penetrate the industry. The effective use of technology can enhance crop yields and address various agricultural issues. Farmers frequently prioritize crops with high market demand and profit potential, overlooking essential aspects like soil health and sustainability, which can result in negative consequences for both agricultural productivity and the environment. When effectively implemented, newer technologies hold the promise of revolutionizing our agricultural sector. The aim seeks to demonstrate the effective application of these technologies in offering valuable support to farmers, with a particular focus on crop recommendation and productivity.

Keywords: Machine Learning, Deep Learning, Crop Productivity, Crop Recommendation, Farming.

Introduction:

In India, farming is not just a business; it deeply influences the social lives of the people involved in it. Numerous festivals and social gatherings are celebrated in alignment with various farming seasons and practices, demonstrating the cultural significance of agriculture[11]. A significant part of the population relies on agriculture, either as a primary source of livelihood or indirectly for sustenance. Yet, the challenges faced by farmers in India remain a pressing issue. The agriculture sector only shares 20% of it to the Indian GDP [1]. [2] This discrepancy underscores the urgent need for improvements to ensure sustainable and profitable agricultural practices without harming the environment. This is where machine learning can play a transformative role[13]. Our paper aims to address the issues faced when farming by recommending the most suitable crops to grow and offering effective solutions to avoid undesirable outcomes, leveraging the power of machine learning techniques.

Literature Review:

In India, farming is more than just a business; it significantly influences the social lives of those involved. Various festivals and social events are closely tied to the seasons and farming activities. Thus, a significant segment of the population relies on agriculture, whether through direct involvement or indirect support. However, Indian farmers face numerous challenges. Despite engaging half of the population [2], agriculture contributes only 20% to the Indian GDP [3]. Therefore, improvements are urgently needed to ensure profitable yields and sustainable practices that do not harm the environment. Technology has the potential to profoundly revolutionize this industry.

This paper seeks to help farmers overcome challenges by providing recommendations on the best crops to grow[4], using advanced machine learning methods to avoid negative impacts. Prior to our research, we reviewed numerous scholarly articles, uncovering various significant findings. These findings are detailed below, with each entry providing the title of the paper, the algorithms used, and the general conclusions drawn from the study.

[1] Ministry of Statistics and Programme Implementation. (2023). Agricultural Statistics at a Glance 2023. It encompasses data on crop production, land use, irrigation, and the economic and social dimensions of agriculture. It provides essential information for policymakers, researchers, and other stakeholders, highlighting key trends and advancements in Indian farming.

[2] World Bank. (2022). India: In 2022, the percentage of total employment in India attributed to agriculture was 42.86%, as reported by the World Bank's collection of development indicators, sourced from official data.

[3] Economic Survey of India. (2023). Sectoral Contributions to Indian GDP - The contributions of the agriculture, industry, and services sectors to the total Gross Value at current prices were 17.7%, 27.6%, and 54.7%.

[4] Rahman, S. A. Z., Islam, S. M. M., & Mitra, K. C. (2020). Soil Classification Using Machine Learning Methods and Crop Suggestion Based on Soil Series - The central idea is to leverage machine learning to streamline soil analysis and provide data-driven crop recommendations. Over the past five years, India's agriculture sector has shown a strong annual growth rate of 4.18%, reflecting an upward trend in productivity and overall output.

[5] Sharma, R., & Singh, A. (2019). Predicting Crop Yields Using Machine Learning Models: A District-Level Approach - To deliver precise and trustworthy crop yield predictions that assist farmers, policymakers, and other key stakeholders in making well-informed decisions.

[6] Sk Al Zaminur Rahman, S.M. Mohidul Islam, Kaushik Chandra Mitra proposed Soil Classification using Machine Learning Methods and Crop Suggestion Based on soil series -A model is designed to predict soil series, recommend suitable crop yields for specific soil types, and suggest appropriate fertilizer use. Additionally, data from other districts will be incorporated to enhance the model's accuracy and reliability.

[7] Sk Al Zaminur Rahman, S.M. Mohidul Islam, Kaushik Chandra Mitra proposed weighted K-NN, SVM, Bagged Tree. SVM has demonstrated superior accuracy in soil classification compared to K-NN and Bagged Tree algorithms, making it an ideal model for categorizing different soil types and recommending appropriate crops for specific regions.

Dataset must have following attributes:

1. Soil Parameters: Type, pH value
2. Climatic Parameters: Humidity, Temperature, Wind and Rainfall
3. Production: Cost of cultivation

In this paper, focus is on predicting crop yields at the district level. Our primary objective is to identify a dataset that includes production details spanning the past 10-12 years, along with information on various climatic parameters such as various compositions[10]. These are crucial for accurate crop prediction using different classification algorithms applied to the dataset. To ensure the effectiveness of our predictions, we thoroughly assess these variables to determine which ones most significantly contribute to accurate crop yield forecasts. The dataset requires preprocessing to address issues such as redundant attributes and noisy data. Initially, we perform data cleaning to identify and exclude redundant factors that do not contribute to crop prediction [4]. During the exploratory data analysis phase, we convert categorical variables into binary values (0 and 1) based on their presence or absence. These binary values facilitate the classification process, enabling us to make more accurate predictions based on each factor.

Proposed system:

The MLP neural networks perform better when increasing in data while traditional Machine Learning algorithms struggle to do so. The neural network is structured with four layers: an input layer, two hidden layers, and an output layer. The input layer contains 10 neurons, corresponding to the 10 input features, while each hidden layer consists of 15 neurons. The output layer has 9 neurons. The Rectified Linear Unit (ReLU) activation function is applied to the hidden layers, and the SoftMax function is used for the output layer. The 'Adam' optimizer is employed for training, with 'Sparse Categorical Cross Entropy' serving as the loss function. The steps to implement the given MLP (Multilayer Perceptron) neural network architecture:

1. Prepare the Dataset:

Collect Data: Ensure your dataset has 10 features as input and 9 output classes.

Preprocess Data: Preprocess the dataset by cleaning it and normalizing the input features to enhance model convergence.

2. Design the Neural Network Architecture: Input layer contains 10 neurons (corresponding to 10 features).The network includes two hidden layers, each containing 15 neurons, and an output layer with 9 neurons, representing the 9 output classes.

3. Choose Activation Functions: The ReLU activation function is applied in the hidden layers, while the SoftMax activation function is used in the output layer to generate probabilities for the nine classes.

4. Compile the Model: Optimizer is for the usage of Adam optimization algorithm for efficient training and loss function is to use Sparse Categorical Cross Entropy since the output involves multiple classes.

5. Train the Neural Network: Set Parameters for batch size (e.g., 32 or 64). Number of epochs (e.g., 50 or until convergence), fit the Model.

6. Evaluate the mechanism: Assess the model's effectiveness.

7. Hyperparameter Tuning: Experiment with various multiple neurons in hidden layers, the rate at which the optimizer adjusts its parameters during training.

8. Deployment: Save the trained model for inference and apply the model to forecast outcomes for new, unobserved data.

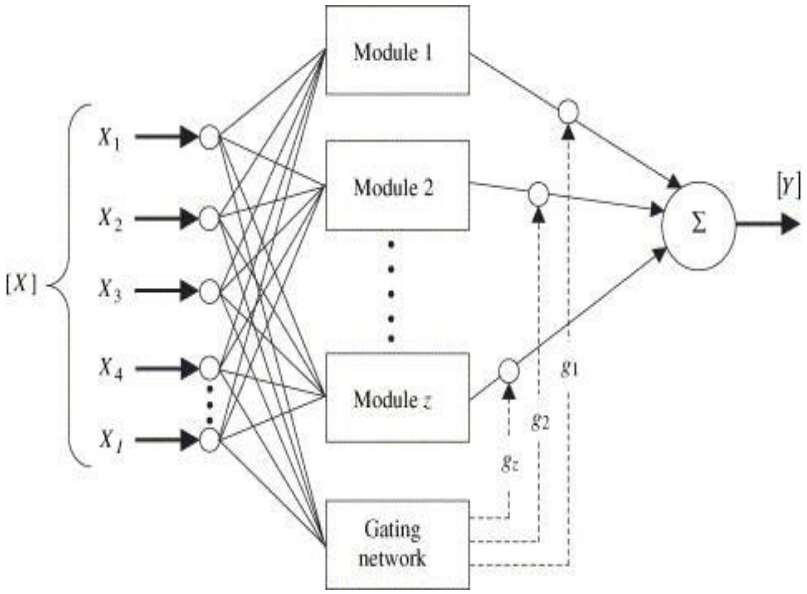

Fig1: MLP neural networks architecture

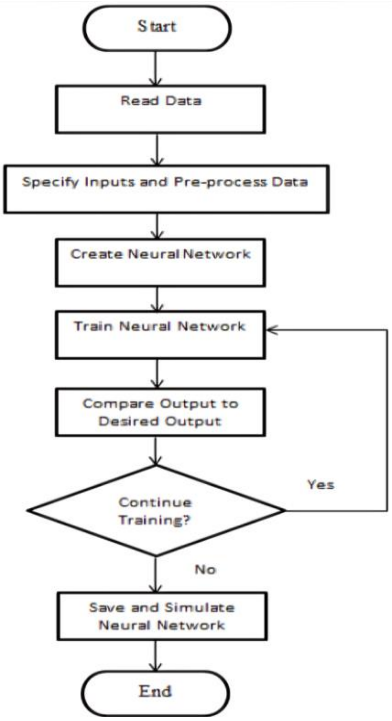

Fig 2: Flow chart of MLP neural networks

The mechanism developed utilizes a neural network to process inputs individually: static soil data with fully-connected layers and dynamic meteorological data with recurrent LSTM (Long Short-Term Memory layers) layers. Trained on historical soil properties, precipitation, and temperature data, the model was tested separately, showing that scalable yield forecasts are possible by detecting redundant information in soil and weather data. It also learns crop cycles from seasonal data, achieving over 75% accuracy across all crops and districts.[15]The user-friendly web platform allows for crop yield predictions based on environmental data and includes features like yield graphs and recommendations for fertilizers and irrigation. The study also explored soil classification [4] using algorithms like J48, BF Tree, OneR, and Naïve Bayes, showing varying accuracy levels [7][18]. Future work aims to expand the data set and incorporate advanced machine learning techniques for better crop yield predictions, potentially boosting agricultural productivity and the economy.[14]Framework testing, or system testing, evaluates a fully integrated system to ensure it meets specified requirements, identifying defects in individual units and the system as a whole[17]. Conducted by an independent testing team, it aims to document system behavior under test conditions objectively. The dataset includes soil-specific attributes from Polytest Laboratories in Pune, Maharashtra, India, and crop data from Marathwada University[8]. The crops analyzed are groundnut, pulses, cotton, vegetables, banana, paddy, sorghum, sugarcane, and coriander. Soil attributes, such as depth, texture, pH, color, permeability, drainage, water holding capacity, and erosion, influence crop growth by affecting water and nutrient absorption, root anchoring, and overall soil health[9]. To improve prediction accuracy [5], an ensemble data mining model using the Majority Voting technique is employed. This method combines the strengths of multiple models, where each model predicts the class independently, and the majority vote determines the final class label. For instance, given specific soil conditions like mildly alkaline pH, deep soil, low water holding capacity, moderate drainage, and low erosion, the system might recommend paddy cultivation.

Methodologies:

The decline in people involved in farming, alongside the growing global population, highlights the urgent need for efficient and precise crop cultivation. This urgency is intensified by climate change and unpredictable weather patterns, posing a threat to food security. To enhance crop yields, innovative technologies must be adopted, bridging the gap between traditional and modern farming practices through software that models climate impacts,

including extreme weather events. Effective climate change adaptation and mitigation strategies require new methodologies and policies. Experimental data helps identify environmental zones [1] affected by weather and water changes, crucial for successful crop cultivation. Weather and pests alter soil types over time, necessitating complex data management. Simplifying reality allows for quick assessment of climate change impacts on agriculture, with models optimizing management practices and crop rotations while implementing new breeding programs. Accurate predictions can track seasonal climate changes, and machine learning software can assess climate impacts and test adaptation scenarios. Data processing analyses long-term experimental data to extract trends, transforming them into actionable information[16]. Machine learning identifies patterns and correlations within large datasets, providing valuable insights for farmers. This information can predict and prevent crop losses, aiding in effective crop cultivation [6].

The libraries used here are Numpy, short for Numerical Python, is an open-source library designed for the Python programming language. Despite the plethora of libraries available, Numpy stands out as one of the most indispensable tools, particularly for numerical computations. Its primary purpose is to support large, multi-dimensional matrices and arrays, complemented by a vast array of high-level mathematical functions tailored for efficient operations on these structures. Numpy simplifies complex tasks such as data transformation and statistical analysis, offering significant speed advantages over native Python functions. For instance, calculating the mean and median of a dataframe can be attained with just a line of code for each operation.

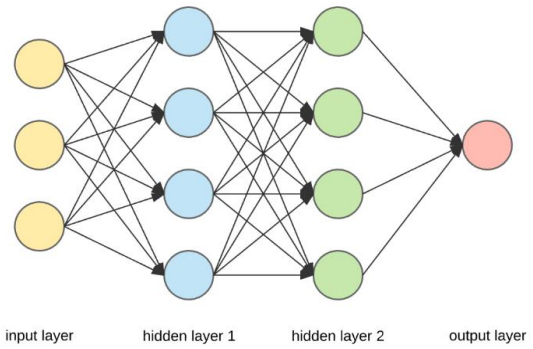

Fig 3: Layers in neural network architecture

Here, we choose the performance metrics listed below to evaluate our model:

- Precision
- Recall

With the metrics we also use these steps for the enhancement of data and various algorithms:

- **Data Cleaning and Preprocessing:**

Ensuring our dataset's accuracy is crucial from the start. Missing values must be filled appropriately, and the normal distribution of features verified. Outliers need addressing to preserve data integrity, and skewed features should undergo normalization. Our dataset's skewed features required quantile transformation for effective normalization.

Parameters for preprocessing are:

Handling missing data, Threshold for missing data, decide on acceptable levels of missingness in rows/columns and imputation strategies. Outlier Detection and Handling. Z-score, IQR, percentile-based filtering, or visualization techniques like box plots. Remove, cap, or transform outliers. Data consistency ensures uniform formats and correct typos and spelling inconsistencies. Duplicate removal identifies duplicates based on key attributes and removes redundant rows or columns. Noise Reduction - apply smoothing techniques. Remove irrelevant features or data points.

Parameters for data cleaning are:

Data scaling/ normalization with range parameters set to custom ranges, feature -engineering consisting of creation, selection, transformation and encoding categorical data with methods as one hot encoding, label encoding, target encoding and binary encoding. Data reduction, data-transformation in dimensionality-reduction.

Log transformations and box corks transformations also handling imbalance data and noise handling.

- **Data Analysis and Visualization:**

After the data has been cleaned and prepared, the subsequent step is thorough analysis and visualization to identify trends and patterns.

Parameters for data analysis visualization are –

Data Parameters are dataset Source -The file or database containing the data. Variables/columns with specific fields or attributes to analyze. Filters with different criteria to subset or segment the data. Data Format is structured. There are also other parameters such as, missing value handling, normalization/scaling, feature selection, outlier treatment, confidence level, threshold values, model parameters, sampling size, processing tools, optimization criteria.

- **Feature Selection:**

To accurately determine the best crop type for cultivation, we selected relevant features using a correlation matrix to assess linear relationships. This matrix helps identify highly correlated features, suggesting redundancy and potential exclusion. Our review shows that the features are not significantly correlated, justifying their inclusion for predicting the optimal crop type.

1. Understand the Data
2. Remove Irrelevant Features
3. Handle Missing Data
4. Assess Feature Importance-filter methods, wrapper methods, embedded methods
5. Evaluate Model Performance
6. Iterate

Formulas:

Mutual Information is I(X;Y)= \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x, y)}{P(x) P(y)}

Feature Importance in Trees FI(f)= \[\sum_{\text{splits on } f} \Delta \text{Impurity}\]

Algorithms:

Random Forest:

Steps:

a) Understand the Data - Identify the target variable and input features.

b) Remove Irrelevant Features - Drop redundant or non-informative features (e.g., IDs, constant columns).

c) Handle Missing Data - Impute or remove features with excessive missing values.

d) Assess Feature Importance - filter methods, tree-based models, etc.

e) Evaluate Model Performance - Test selected features using a baseline model.

f) Iterate - Refine based on evaluation metrics to balance model simplicity and performance.

Example pseudocode:

Input: A training dataset DD consisting of features XX and target YY, with TT decision trees and mm features selected for each split.

Output:

1. Create an empty list to store the decision trees: Forest = []

2. For each tree t in range(T):

a. Create a bootstrap sample D_t by randomly sampling from D with replacement.

b. Build a decision tree:

i. At each node:

- Arbitrarily select m features from the total feature set.

- Identify the optimal split using the chosen m features.

ii. Grow tree until a stop condition occurs

- Add this trained tree to Forest.

3. Return Forest

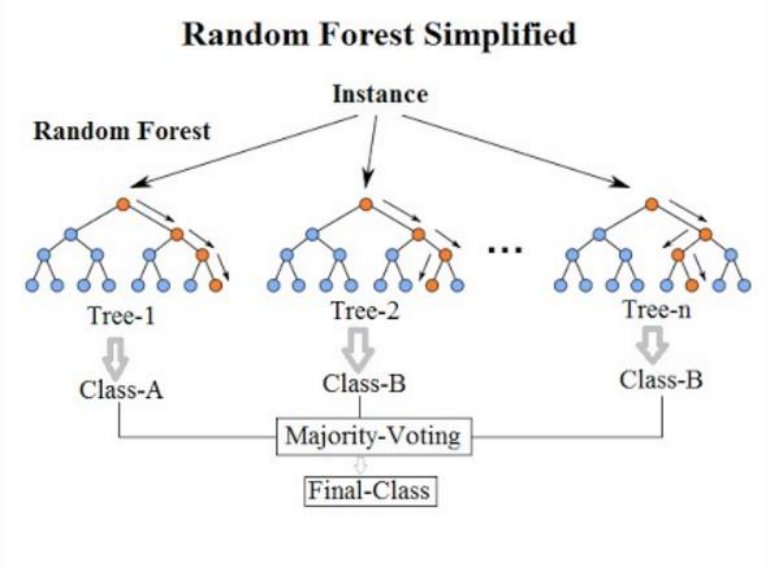

Fig 4: Random Forest

Decision Tree:

Steps:

a) Start with the root node by encompassing the entire dataset.

b) Choose the optimal split - by determining the feature and threshold that best divide the data to increase information gain or reduce impurity.

c) Split the Data - Partition the dataset into subsets by splitting it according to the selected feature and threshold.

d) Create Child Nodes - Assign subsets to child nodes and repeat the splitting process for each child.

e) Stop Splitting - Stop when a stopping criterion is met.

f) Assign Leaf Nodes - Label leaf nodes with the majority class or the average value.

g) Tree is Ready - Use the tree to make predictions by traversing it from root to leaf based on feature values.

Example Pseudocode:

Input: Training dataset D with features F and target variable Y

Output: A decision tree T

1. If every instance in DD is assigned to the same class or has an identical target value:

- Return a leaf node corresponding to the given class or target value.

2. If F is empty (If no features remain for splitting, return a leaf node with the dominant class (for classification) or the mean target value (for regression) in D.

3. For classification and regression:

- For classification: Use a metric like Gini Index, Entropy (Information Gain), or Gain Ratio.

-For regression: Use a metric like Variance Reduction or Mean Squared Error (MSE).

4. Divide the dataset D into smaller subsets with F.

5. Create a decision node for F_best:

- For each subset Di:

- Recursively call the algorithm on Di with remaining features F - {F_best}.

- Attach the resulting subtree to the decision node.

6. Return the constructed tree T.

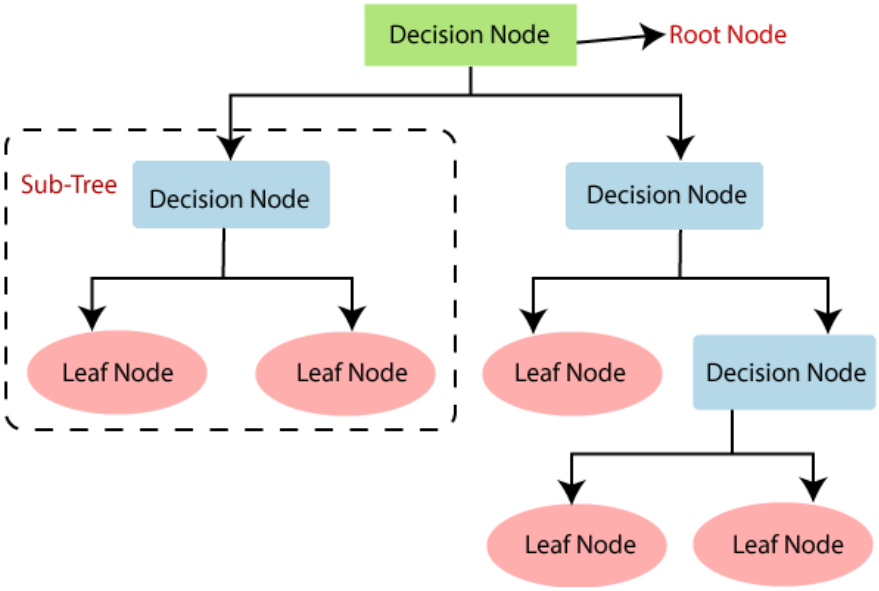

Fig 5: Decision Tree

Logistic Regression:

Steps:

a) Start with the root node starts by using the complete dataset.

b) Select the Best Split - Identify the feature and threshold that split the data to maximize information gain or minimize impurity.

c) Split the Data - The data is processed to optimize information gain or reduce impurity and threshold.

d) Create Child Nodes - Assign subsets to child nodes and repeat the splitting process for each child.

e) Stop Splitting - Stop when a stopping criterion is met.

f) Assign Leaf Nodes - Label leaf nodes with the majority class or the average value.

g) Tree is Ready - Use the tree to make predictions by traversing it from root to leaf based on feature values.

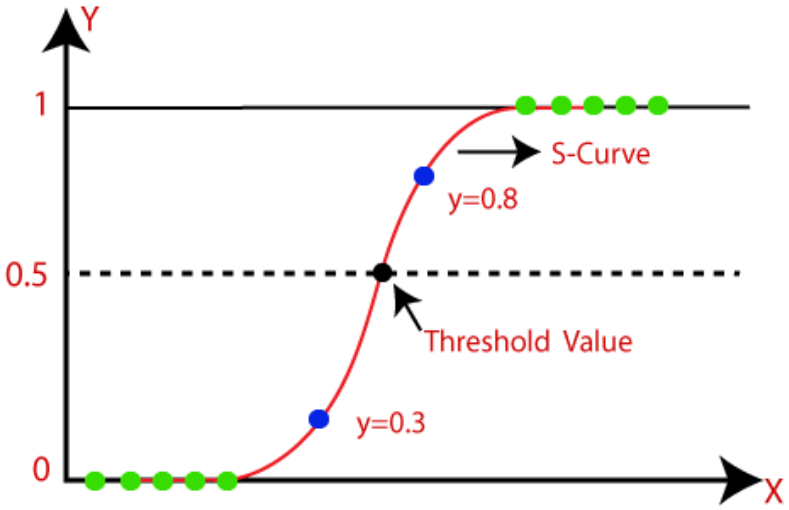

Fig 6: Logistic Regression

Formulas:

The sigmoid function transforms any real-valued number to a value in the range of 0 and 1:

\sigma(z) = \frac{1}{1 + e^{-z}}

Compute gradients for weights w and bias b:

For the gradient with respect to *w:*

\frac{\partial \text{Loss}}{\partial w} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i

For the gradient with respect to *b:*

\frac{\partial \text{Loss}}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)

Update weights and bias using gradient descent:

For updating *w:*

w = w - \alpha \cdot \frac{\partial \text{Loss}}{\partial w}

For updating *b:*

b = b - \alpha \cdot \frac{\partial \text{Loss}}{\partial b}

XG Boost:

Steps:

a) Initialize Model - Start with an initial prediction.

b) Compute Residuals - Calculate the residuals based on the current predictions.

c) Build Decision Trees - Fit a decision tree to predict the residuals. Apply regularization to prevent overfitting.

d) Update Predictions - Add the weighted output of the new tree to the previous predictions. Use a learning rate to scale the contribution of the tree.

e) Repeat - Iterate steps 2–4 for a predefined number of trees or until convergence.

f) Final Prediction - Combine predictions from all trees to generate the final output.

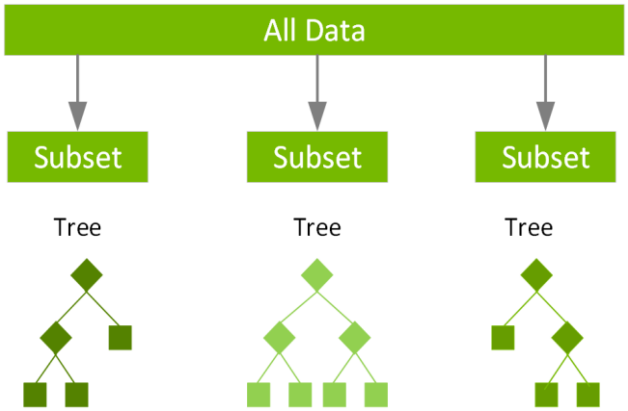

Fig 7: XGBoost

Example Pseudocode:

Input: Training dataset D = {(xi, yi)} for i = 1 to n

Learning rate η, number of trees T, loss function L(yi, ŷi)

1. Initialize model:

$F_0(x)$ = constant

2. For t = 1 to T (number of trees):

a. Compute pseudo-residuals:

$r_i = -[\partial L(yi, F(xi)) / \partial F(xi)]$ for all i = 1 to n

(Gradient of the loss function w.r.t. current prediction)

b. Fit a base learner to predict the residuals:

Train tree $h_t(x)$ on $\{xi, r_i\}$

c. Compute leaf weights for the tree:

For each leaf j in the tree:

$w_j = \sum(r_i) / (\sum(\text{hessian of L for xi in leaf j}) + \lambda)$

d. Update the model:

$F_t(x) = F_{t-1}(x) + \eta * h_t(x)$

3. Output the final model:

$F(x) = F_0(x) + \Sigma(\eta * h_t(x))$ for t = 1 to T

Results:

RF's Accuracy is: 0.990909090909091

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.94	1.00	0.97	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.90	1.00	0.95	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	0.95	0.97	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.81	0.90	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Fig 8: Random Forest algorithm accuracy

The image presents a table displaying the performance metrics of various crops evaluated using a Random Forest algorithm. Each crop is assessed with various values. The overall accuracy of the mechanism is reported to be approximately 99%, with both macro and weighted averages also reflecting this high level of accuracy across the different categories.

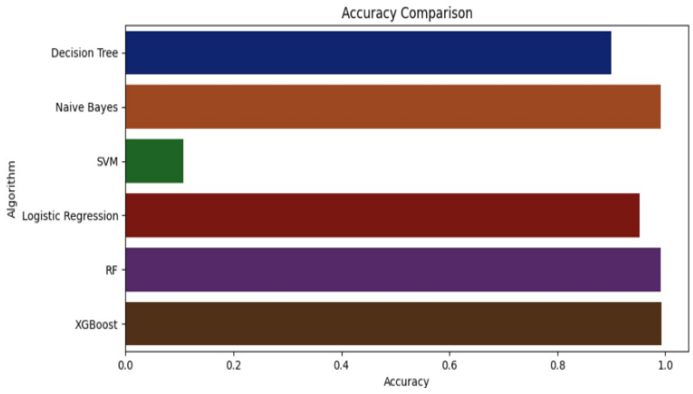

Fig 9: Accuracy comparison of all algorithms shown in horizontal bar chart

The above image is a horizontal bar chart displaying the performance evaluation of various algorithms. The y-axis lists the algorithms while the x-axis represents accuracy values ranging from 0.0 to 1.0. This chart visually highlights how each algorithm performs in terms of accuracy.

Model	Accuracy(approx.)
Random Forest	0.9909
SVM	0.10681
Logistic Regression	0.95227
Decision Tree	0.9
Naive Bayes	0.9909
XGBoost	0.98318

Table 1: Approximate accuracies of algorithms

The above table displays the approximate accuracies of various machine learning algorithms, with their corresponding accuracy values. For example, both Random Forest and Naïve Bayes show an accuracy of approximately 0.99090. The table is designed in a clear format with numerical values and labels for easy interpretation.

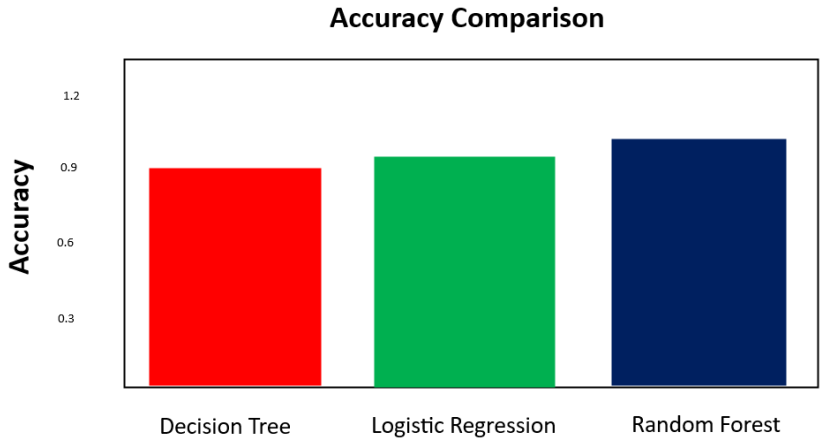

Fig 10: A vertical bar chart illustrating the accuracy comparison between decision tree, logistic regression, and random forest models.

The chart visually represents the accuracy values associated with each model using bars. It captures data in a clear design, incorporating numerical values and distinct colors for each model to enhance readability. The outcomes are derived from the present dataset sourced from publicly available resources. These can vary depending on many factors involved.

Conclusion:

This research developed an advanced crop productivity system to improve farming practices, providing tailored crop suggestions based on various factors.

The AgroSysNN neural network model achieved an impressive 99.7% accuracy and 0.003 loss, outperforming traditional algorithms. Future work involves enhancing the model with larger datasets and real-world applications. Comparative analyses of classifiers using crop-related variables were visualized with confusion matrices, revealing the effectiveness of decision tree and other algorithms[2]. The study also aims to create a web-based solution using a Multilayer Perceptron Neural Network for crop recommendations to boost yield and profitability.

References:

[1] Government of India, Ministry of Agriculture and Farmers Welfare, Agriculture Census 2021. Provides detailed data on agriculture in India, including the social and economic implications of farming - agriculture.gov.in .

[2] Indian Council of Agricultural Research (ICAR), State of Indian Agriculture Report (2021-22). An official report discussing the current state of agriculture in India and areas needing technological intervention - icar.org.in .

[3] Jain, A., Singh, P., & Kumar, R. (2021), Machine Learning Approaches in Agriculture: A Survey discusses the role of machine learning techniques in enhancing agricultural productivity and sustainability.

[4] Das, N., & Singh, S. (2020), "Socioeconomic Impacts of Agriculture in Rural India: Challenges and Solutions." examines the relationship between agriculture, social life, and the economy in rural India - Journal of Rural Development.

[5] Pradhan, A., & Mishra, B. (2019), "Crop Recommendation Using Machine Learning Techniques." explores the use of algorithms like Random Forest and SVM for crop recommendation based on environmental factors - International Journal of Computer Applications.

[6] International Conference on Smart Agriculture and Environment (ICSAE 2022). Topics include AI and machine learning applications in agriculture, with case studies specific to India.

[7] National Seminar on Sustainable Agriculture Practices in India (2021). Hosted by ICAR, this seminar discusses policy recommendations and the role of technology in Indian farming.

[8] Food and Agriculture Organization (FAO), "The Role of Agriculture in Indian Society." Discusses agriculture's cultural and economic significance in India - www.fao.org .

[9] World Bank, "Agriculture in South Asia: Challenges and Opportunities." offers insights into the economic and technological aspects of agriculture in the region - www.worldbank.org .

[10] Government of India, Ministry of Agriculture & Farmers Welfare: "Annual Report 2022-23". Provides comprehensive statistics and insights into the agricultural sector in India - agricoop.nic.in .

[11] National Sample Survey Office (NSSO), "Situation Assessment of Agricultural Households and Land Holdings of Households in Rural India". Highlights the economic status and challenges faced by Indian farmers - mospi.nic.in .

[12] NITI Aayog Report: "Doubling Farmers' Income by 2022". Discusses strategies and challenges for improving farmer incomes and productivity - niti.gov.in .

[13] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018): "Deep learning in agriculture: A survey". Computers and Electronics in Agriculture, 147, 70-90. This paper provides an overview of deep learning techniques in agricultural applications.

[14] Patil, R. R., & Biradar, N. (2021): "Application of machine learning techniques for precision agriculture: A review". Agricultural Reviews, 42(3), 250-257. Focuses on the use of ML algorithms for precision farming.

[15] Lobell, D. B., Schlenker, W., & Costa-Roberts, J. (2011): "Climate trends and global crop production since 1980". Science, 333(6042), 616-620. Explores how climate variability impacts agricultural productivity globally.

[16] Chollet, F. (2018): "Deep Learning with Python". Manning Publications. Provides an introduction to neural network architectures like MLP and LSTM.

[17] Kumari, R., & Pandey, M. (2020): "Crop prediction using machine learning algorithms". International Journal of Engineering Research & Technology (IJERT), 9(2), 123-126. Discusses specific ML techniques like Random Forest and Decision Trees for crop yield prediction.

[18] ICAR – Indian Council of Agricultural Research: "Vision 2050" and other reports detailing soil health, climate impacts, and agricultural practices - icar.org.in .