International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Region-Wise Weather Data-Based Crop Recommendation System Using Different Machine Learning Algorithms

Saikat Banerjee[1]*, Dr. Abhoy Chand Mondal[2]

Submitted: 25/04/2023 **Revised**: 24/06/2023 **Accepted**: 06/07/2023

Abstract: In India, the majority of people make their living from the agricultural sector. Agriculture provides a livelihood for approximately 50 percent of India's total population. Agriculture has a significant bearing on both the state of business and the level of food safety in the country. This part of the economy is in terrible shape and has been showing signs of stagnation for some time. Changes in the environment make it harder for farmers to grow enough food for everyone. Floods and droughts are just two examples of bad weather events that can change the growing season, reduce the amount of water available, help plants, bugs, and fungi spread, and, in the end, make farming less productive. Greenhouse gas pollution from farms can be reduced if farmers adopt climate-smart practices. Farmers who employ AI find it easier to make sense of environmental and operational variables like temperature, humidity, wind speed, and sun radiation. Collecting and assessing statistics on climate, precipitation, soil, seed, crop yields, humidity, and wind speed in a few key locations will help producers increase crop yields. In addition, a custom recommender system was used to forecast crops and display them on a Flask-built graphical user interface. The system's flexible architecture means it could one day be used to find the recommended products of other regions. This article develops similar machine learning methods and compares and contrasts five methods using specific agricultural data.

Keywords*: Decision Tree, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machines*.

1. Introduction

Agriculture is the central way that the vast majority of people in India make a living, so the industry should never be taken for granted. Despite the fact that agriculture's endowment to the comprehensive gross domestic product (GDP) now ac-counts for less than 20% and that the contributions of other industries have grown at a faster rate, agricultural production has gone up. We have achieved self-sufficiency as a result of this, and since gaining our independence, we have shifted from being a food centre to becoming a positive producer of agricultural commodities and other associated products [12]. It is anticipated that the total production of foodgrains across the country will hit a new high of 291.95 million metric tonnes in 2019–20, as stated in the second advance prediction for that time period. Nevertheless, the Indian Council of Agri-cultural Research (ICAR) has estimated that the demand for foodgrains will increase to 345 million metric tonnes by the year 2030. India is fortunate to have a large quantity of fertile territory because the Indian Council of Agricultural Research (ICAR) has defined 15 agroclimatic zones that encompass the country's diverse range of weather conditions and soil types [14, 16]. These zones allow India to grow a diverse range of products. India comes in second in the production of rice, wheat, oilseeds, fruits, vegetables, sugarcane, and cotton [23]. However, the country ranks first in the production of milk, seasonings, legumes, tea, almonds, and jute [24]. Agriculture market changes were impacted more quickly by policies towards privatisation, lib-eralisation, and globalisation [8]. Agricultural marketing reforms put in place after 2003 changed how agricultural products were sold by letting private investors set up markets, contract farms, trade futures, and do other things [11]. Even though the rate of change is lower, these changes to marketing practises have led to some changes. Young people with good educations have started a lot of businesses in agriculture, which shows that they can see the huge potential of investing money and time in this industry [2]. Because of the cumulative effects of technology, agriculture will change in the next ten years. Other things, like the information technology revolution in India, new agricultural technologies, private investments, especially in research and development, government efforts to revive the cooperative movement to deal with the problems of small holdings and small produce, etc., are also changing the face of agriculture in India [9, 10]. Despite these realities, average production levels for a number of crops in India are below par [12]. The country's population is expected to increase to the largest in the world in the next decade, making food security a major concern [22]. Farmers still have a hard time making ends meet. As India's population, median income, and the impact of globalisation all grow, so too will the need for a diverse variety of foods [19]. Therefore, there will always be a de-

1 State Aided College Teacher, The department of Computer applications, Vivekananda Mahavidyalaya, Haripal, Hooghly, West Bengal, India, Email: Saikat.banerjee56@gmail.com, ORCID ID:https://orcid.org/0000-0002-7361-1553

2 Professor and Head, The department of Computer science, The University of Burdwan, Golapbag, West Bengal, India, E-mail:acmondal@cs.buruniv.ac.in,

** Corresponding Author Email: Saikat.banerjee56@gmail.com*

mand for additional crop production in terms of quantity, variety, and quality [25], despite the fact that agricultural land is limited. Machine learning (ML), a subfield of artificial intelligence, uses statis-tical methods to give computers the ability to get better as they are used more. Computers and computer-controlled machines can now take action and make data-driven decisions without being strictly designed to do so [13]. These days, most applications of ML focus on either classifying data with pre-trained ML models or making predictions about the future. ML refers to the steps involved in care-fully collecting features [8]. Supervised learning, uncontrolled learning, and reinforcement learning are the three main kinds of methods used in ML. Supervised learning, wherein marked datasets are used for training, is essential for the development of algorithms that can effectively categorise data or forecast outcomes [15]. Using marked inputs and outputs, the model's precision can be evaluated, and it can be trained over time. Simply stated, we use the training input and the desired output to train the machine, and then we use the test datasets to make predictions about the trained machine's performance. Supervised learning primarily addresses classification and regression problems [1]. In the present day, ML is mostly used to do two things: classify data with pre-trained ML models and make predictions about the future. Supervised learning is used to teach algorithms that aid in accurate categorization of data or prediction of results by utilising annotated datasets [4]. The accuracy of the model can be assessed with the help of marked inputs and outputs, and the model itself may change over time. To put it simply, we first provide the computer with training input and the intended output, and then we use test datasets to make predictions about the training dataset [14]. Classification and recurrence issues are the two mainstays of guided learning. Mainstream applications of ML today involve classifying data according to predefined categories and making pre-dictions about the future with the help of prebuilt models. To address the categorization problem, we employ a classification strategy that generates a category response variable [17]. Real-world applications include spam filtering, voice recognition, handwriting recognition, document classification, fingerprint authentication, and many more. Com-mon classification strategies include random forest, decision tree, logistic regression, support vector machine (SVM), and others [19]. Regression al-lows for forecasting by allowing us to find the relationship between some characteristics of our data and some observed continuous-valued reactions [15]. The market, the weather, and other constant factors can all be predicted with this method. Regression analysis makes use of well-known methods like simple linear regression, multivariate regression, decision trees, and Lassa regression [16]. There is no relationship between the input and out-come variables in unsupervised learning. Modelling the underlying structure or distribution of the data is the main goal of the technique used to identify or categorise the unnamed datasets based on patterns, parallels, or variations [6]. When it comes to autonomous learning, clustering and linkage fall into separate groups. During clustering, dissimilar data is grouped together, while similar data is grouped together. This monetary-spending-based segmentation method is employed by businesses to better serve their clients [7]. Some examples of clustering methods are K-means clustering, the mean-shift approach, DBSCAN, PCA, ICA, and many others. In reinforcement learning, we use a feedback-based method [21]. It allows software workers to make their own decisions about the best actions to take in any given circumstance, increasing productivity as a result [18]. The foundation of reinforcement learning is the dynamic interplay between the earn-ing agent and its surrounding world. Here, the individual benefits from positive reinforcement for good behaviour and negative reinforcement for bad behaviour [19, 20]. Agents are provided feedback in the form of praise and punishment to help them improve [9]. Similarities between the reinforcement learning process and human learning can be seen in the way a child picks up knowledge from everyday experiences [10]. There are two types of reinforcement learning, each with its own advantages and disadvantages [12]. When an occurrence happens because of a certain behaviour, positive reinforcement learning enhances and increases the regularity of the behaviour [18]. Negative reinforcement learning strengthens behaviours by reducing their likelihood of failure. Reinforcement learning has numerous applications, including but not limited to video games, resource management, automation, text extraction, and many others [26]. The ultimate goal of the research is to identify indicators of field variability that are optimal for use in yield suggestion tasks. Recommending output from previous and in-season exorcisms is the subtask here. In addition, from a farming perspective, it will be fascinating to see how much of an impact fertilisation has on output in the present site year. It is possible to use ML algorithm methods for this goal, but they need to be investigated first. Finding suitable data models that are both accurate and ubiquitous in their return suggestion competence is, therefore, the focus of this study. For this purpose, multiple ML methods will be put to the test using the exact same datasets. This article aims at presenting a new web-based strategy of ML for addressing the farm-ing issue.

2. Literature Review

In all of these models, temperature, rainfall, and soil type were found to be the most common parameters [1], and artificial neural networks were found to be the most common method. Analysis indicates that the three types of deep learning algorithms—Convolutional Neural Networks, Long-Short Term Memory, and Deep Neural Net-works—are the ones that are utilised most frequently in this

investigation.

The author of [2] took things like the state, district, season, and region into account. As a result, the user can predict agricultural production in any year. The study improves the efficiency of the algorithms and makes an accurate forecast of the output by using complicated regression techniques

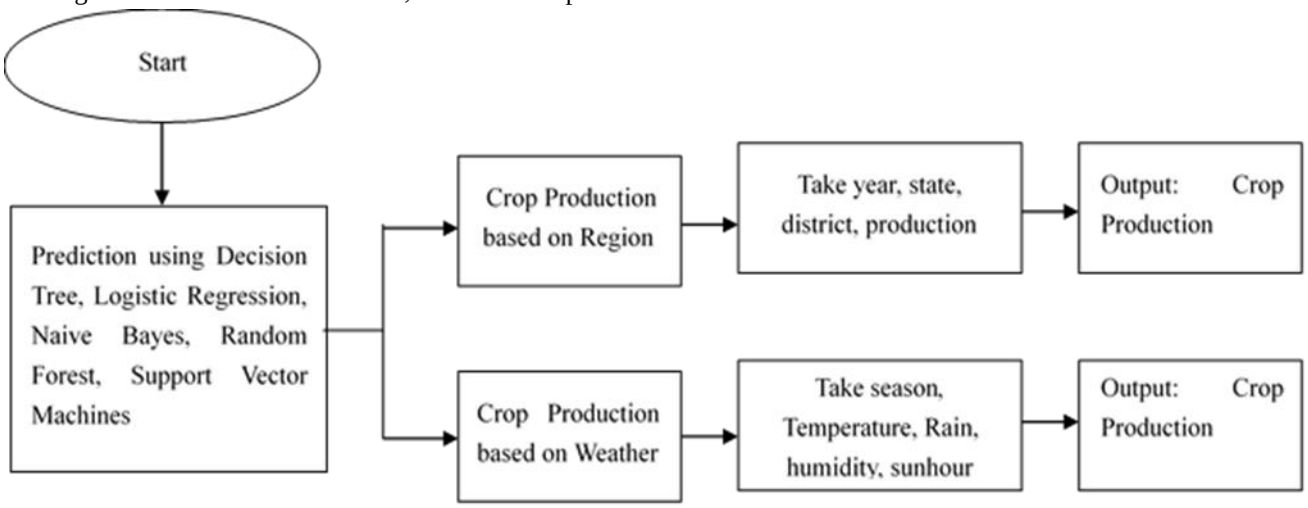

Fig. 1. Flowchart of proposed system

such as the , Lasso, Kernel Ridge and ENet algorithms, and the concept of layered regression. The effectiveness of this effort is being evaluated using the root mean square error.

Article 3 says that agricultural yield prediction involves figuring out how much a crop will produce by looking at historical data like temperature, humidity, pH, rainfall, and crop name. Other factors that are considered in agricultural yield prediction include the crop name. Data mining strategies and random forest ML algorithms were utilised by the author in order to make predictions regarding agricultural yields. The technique that is being suggested can be of assistance to producers in getting a better understanding of the desire for and sale of different products. It provides guidance to producers so that they can select the most profitable product to grow in their fields. [13,18]

The author of [4] says that when training the random forest classifier, you should use forward feature selection and hyperparameter modification. This strategy is accurate to a 71.88% degree. The precision of this tactic was evaluated in comparison to the accuracy of two conventional learning benchmarks. The actual output from the preceding year is going to serve as the original benchmark for the prediction. The second criterion is that an experienced person is responsible for making an accurate projection of the expected harvest from each area. [8,11]

The person who wrote this study [5] says that data mining and ML techniques can help farmers choose crops and predict agricultural production. Data mining is the process of finding a new pattern in a lot of old information. This pattern is used to get information that helps farmers choose crops based on a number of easily accessible traits and also to calculate the output of agriculture. The au-thor made predictions about the data by utilising the linear regression method and then contrasted those predictions with those made using K Nearest Neighbour. It has been demonstrated that the linear regression methodology is more accurate than the K-NN approach when it comes to the selection of agricultural varieties [9, 10],

The author of [6] says that agricultural prediction depends on things like temperature, land productivity, amount of water, quality of water, seasons, the price of produce, and many other things. By taking into account location, weather data, and time of year, ML can accurately predict agricultural outputs. It gives help to farmers so that they can grow the things that are best for the area they farm in. The author evaluated the prediction system by using ML techniques such as SVM, random forest, and Iterative Dichotomize 3, and he did so by utilising historical data. The SVM provides the most accurate results.

In [7], the process of meta-learning was utilised to forecast the values of selected significant commodities over the course of time. Crop pricing and crop production databases are trained using this hybrid of a self-organized map and a long-short-term model. One use of this melange is in the field of adaptive crop price prediction using machine learning. The results of the experiments show that the existing agricultural price prediction systems can be significantly improved in terms of both their precision and their degree of cross-correlation entropy.

3. Proposed Methodology

Our proposal entails the development of a crop recommendation tool that utilises weather statistics and employs advanced ML algorithms. The resulting application will be web-based and will account for crucial parameters, including temperature, rainfall, humidity, and solar hours, to predict optimal vegetation. The proposed system's step-by-

step workflow is depicted in Figure 1. Broadly speaking, we are making three prognostications: first, we envisage the growth of plants based on their seasonal production; second, we forecast the weather patterns for a specific month; and third, we predict the yield of crops based on climatological information. Subsequently, relying solely on meteorological data, it is possible to draw a comparison between our agricultural yield and the seasonal output. The enhancement of the audience's livelihoods can be achieved through di-verse means. However, our objective was to employ statistical analysis to forecast a beneficial outcome, thereby enabling resource managers and farmers to make informed decisions regarding planning. The proposed system architecture out-lined above represents a systematic approach employed in the field of system analysis with the aim of identifying, clarifying, and organising system requirements. A use case is a set of hypothetical interactions between people and machines, carried out in a particular setting and with a particular goal in mind. The use of a use case document can facilitate the identification and comprehension of potential transactional errors by the development team, thereby enabling them to effectively resolve such issues. A conceptual mod-el known as the system architecture is required to delineate the structure and conduct of a system. The formal representation of a system is inherent within the system itself. The term "system architecture" can encompass either a conceptual framework utilised to elucidate the system or a methodology for constructing it, contingent upon the particular context. It is beneficial to analyse research early on by creating an appropriate system architecture. The diagram depicted in Figure 2 il-lustrates the proposed methodology for a crop recommendation system.

3.1 Data Analysis

Assessing information is among the primary activities carried out during the implementation process. This study was conducted with our assistance in an effort to identify potential correlations among the diverse attributes present in the datasets. Ac-quiring knowledge and skills through formal instruction and learning experiences. Datasets: The efficacy of a machine's ability to learn a given set of rules is contingent upon the quantity of parameters involved and the accuracy of the training dataset. The present study involved an examination of multiple datasets sourced from government websites and private website. A meticulous selection process was employed to identify the criteria that were deemed likely to yield favourable out-comes. Numerous studies in this domain have examined environmental indicators as a means of forecasting agricultural sustainability. Certain investigations have employed yield as the principal variable, whereas others have relied on basic eco-nomic factors. Numerous studies in this domain have examined environmental indicators to fore-cast agricultural sustainability. In India, West Bengal agriculture has used around 3% of the country's arable land. The West Bengali agriculture industry produces more than 8% of the food consumed in India. We consider major crops in West Bengal, India, when preparing the dataset. Rice is widely recognised as the primary staple crop in the state of West Bengal. Additional prominent agricultural commodities comprise maize, wheat, mesta, urad, soybean, moong, masoor, groundnut, sugarcane, sunflower, peas and beans (pulses), and jute. We have included these 13 major

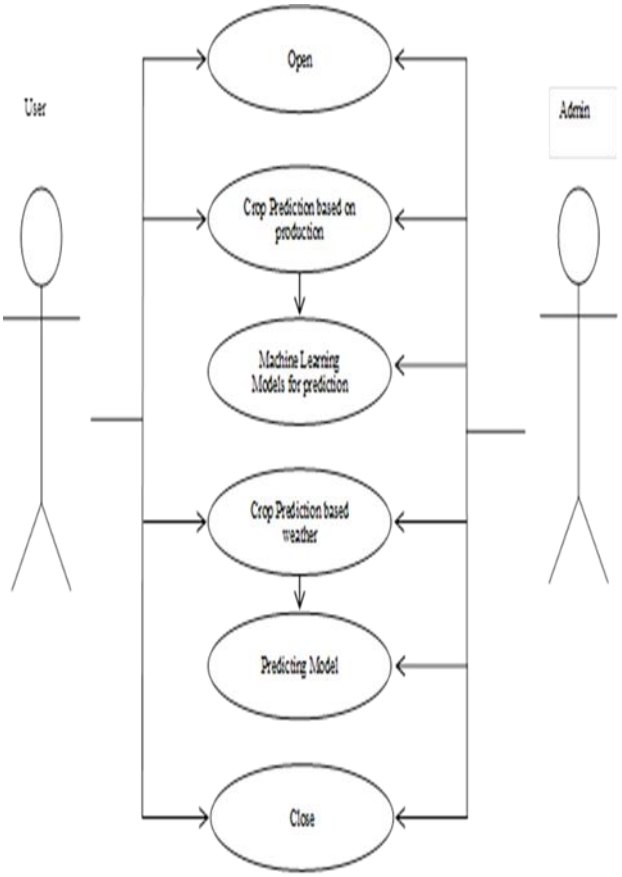

Fig. 2. Flowchart of system architecture

crops in our dataset. We are collect-ing data from two districts: Purba Barddhaman and Hooghly. The district of Purba Barddhaman heavily relies on rice cultivation, as it is the primary crop grown in the alluvial plains situated towards the east with minimal cultivation of other crops. Rice cultivation encompasses a diverse range of varieties that can be broadly classified into three primary classes based on distinct characteristics. These classes include the Aus or autumn, the Aman or winter, and the Boro or summer rice. Paddy cultivation encompasses a significant proportion of the total cultivated land. Potato, oil seeds, jute, mesta, and sugarcane are commercially cultivated crops in marginal areas [24, 27]. The primary crops of the Hooghly district are rice and potatoes. Despite the district's propensity for drought, it may increase food production when there is enough rainfall. Other important crops include wheat, vegetables, mustard, potatoes, and rice. Similar to rice, the area produces an abundance of vegetables and potatoes.

With the introduction of new types of pulse crops including moong, masoor, and oilseeds, the district is falling behind in the production of pulses. To fill the gap between the demand for and supply of oilseed crops, groundnut and sunflower were introduced during the Rabi season [27]. The study aimed to integrate environmental factors, including rainfall, temperature, humidity, and sun exposure, with economic considerations such as manufacturing and proximity to provide reliable and precise recommendations to farmers regarding the most suitable crop for their land. Table 1 displays a sub-set of the ultimate dataset, which comprises 5809 rows and 10 columns. The dataset should be partitioned into two distinct sets, one for training and the other for testing. The training set should com-prise 80% of the data, while the remaining 20% should be allocated to the testing set for the purpose of model evaluation.

3.2 Data Preprocessing

The preprocessing stage is typically performed subsequent to the analysis and visualisation of data. The process of data preprocessing is of utmost importance as it involves the cleansing of data and its preparation for utilisation in ML algorithms. Pre-processing primarily focuses on eliminating outliers or erroneous data and handling missing values. There exist two distinct methodologies for addressing the issue of incomplete data. One possible approach is to remove the entire row that contains the erroneous or absent data. Although the implementation of this approach is straight-forward, it is advisable to apply it solely on extensive datasets. The implementation of this approach on small datasets may lead to an inadequate amount of data, particularly in cases where there is a high incidence of missing values. The potential implications of this could be substantial in terms of the precision of the final result.

Table. 1. Dataset

State Name	*District Name*	*Crop Year*	*Season*	*Crop*	*Production(thousand tons)*	*Temparetue©*	*Rain fall(mm)*	*Humidity(g.m-^{3})*	*Sun hours(W/m^2)*
West Bengal	PURBA BAR DHAMAN	2020	Kharif	Soyabean	13	27.1	1183	80.2	8.3
West Bengal	HOOGHLY	2017	Kharif	Mesta	1699	28.1	1326	82.3	8.4
West Bengal	PURBA BARDHAMAN	2021	Summer	Rice	110708	29.7	236	70.5	9.7
West Bengal	PURBA BAR DHAMAN	2020	Kharif	Urad	154	27.1	1183	80.2	8.3
West Bengal	PURBA BAR DHAMAN	2021	Annual	Wheat	383	26.4	1501	79.1	9.2
West Bengal	HOOGHLY	2017	Summer	Rice	313613	30.2	253	66.4	9.6
West Bengal	HOOGHLY	2014	Rabi	Potato	3434459	22.7	231	69.1	9.2
West Bengal	PURBA BAR DHAMAN	2010	Rabi	Masoor	614	21.3	165	63	9.1
West Bengal	PURBA BAR DHAMAN	2020	Kharif	Sunflower	249	27.1	1183	80.2	8.3
West Bengal	HOOGHLY	2017	Rabi	Peas & beans (Pulses)	154	22.9	218	68	9.2
West Bengal	BARDHAMAN	2018	Summer	Groundnut	1111	34.3	216	53	9.4
West Bengal	HOOGHLY	2018	Kharif	Jute	516911	25.9	1429	83	8.1
West Bengal	PURBA BAR DHAMAN	2020	Summer	Moong	906	32.8	221	59	9.8
West Bengal	HOOGHLY	2020	Annual	Sugarcane	46862	26.8	1769	83	8.9

4. ML Models

4.1 Decision Tree

The decision tree is a diagram that is drawn in the style of a flowchart. The tree's internal nodes stand for characteristics, its branches for decision-making rules, and its leaves for the outcome. In a decision tree, the root node is the starting point. Given a value for an attribute, the algorithm learns to partition the data accordingly. The tree is partitioned through a recursive process known as recursive partitioning. The diagrammatic representation resembling a flowchart is a useful tool for facilitating the process of decision-making. Visualisation in the form of a flowchart diagram is capable of emulating human cognitive processes with ease. Decision trees are easily comprehensible and interpretable, which is why they are preferred. The Decision Tree algorithm is classified as a transparent ML model. The internal decision-making process, which may not be discernible within the opaque algorithmic structure of neural networks, is disclosed. The educational process exhibits a higher rate of efficiency in comparison to the neural network algorithm. Decision trees

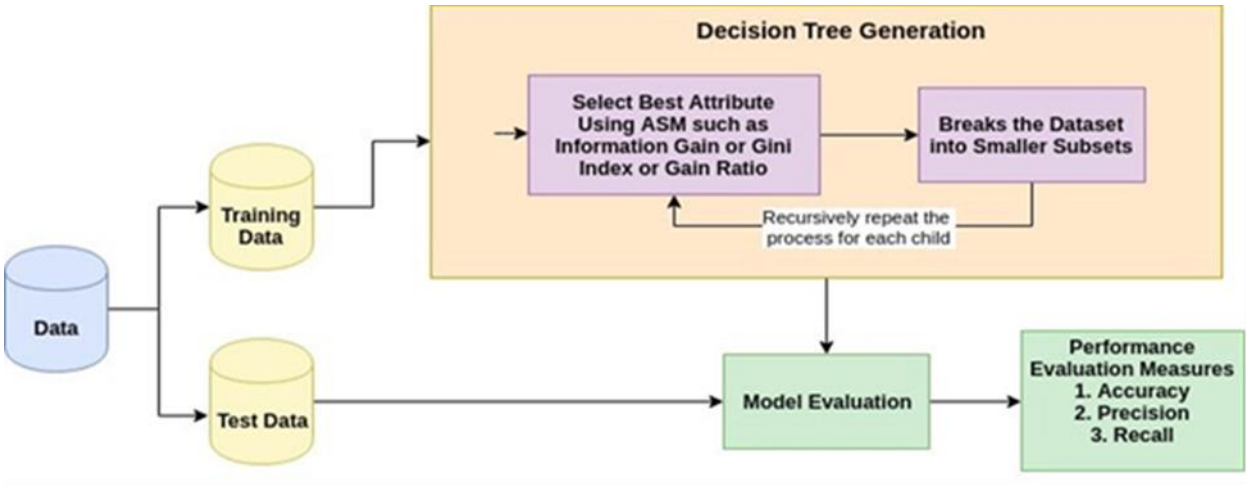

Fig. 3. Decision Tree Workflow

' execution times vary according on dataset characteristics and the number of available attributes. The decision tree is a statistical method that is regarded as being non-parametric or distribution-free. This is because the selection tree does not depend on any fundamental beliefs about the probability distribution. Decision trees have the capability to effectively handle high-dimensional data with appropriate precision. The workflow of the decision tree is illustrated in Figure 3.

The following is an explanation of the idea behind the decision tree algorithm:

1. One can employ attribute selection measures (ASM) to identify the most suitable attribute for dividing the records.
2. The attribute mentioned above should be transformed into a decision node, and subsequently, the dataset must be partitioned into smaller subsets.
3. The procedure of constructing a tree commences with an initial step that is repeated iteratively for every child until one of the subsequent conditions is satisfied.

All the tuples are related to a single attribute value. All attributes have been exhausted. There are no further occurrences.

4.1.1 Metrics for Selecting Attributes

Attribute selection measures are heuristics for determining which data-splitting criteria are optimal. The technique referred to as splitting rules is utilized to identify the breakpoints for tuples on a specific node. The given dataset is explained by ASM through the provision of a rank for each feature. The splitting attribute will be selected based on the best score attribute. When dealing with a continuous-valued attribute, it is necessary to define split points for branches. The commonly used selection measures include information gain, gain ratio, and Gini index

4.1.2 Information Gain

Decision Tree

```
In [15]: from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(Ytest,predicted_values))
```

Fig. 4. Decision Tree Workflow code

Shannon came up with the idea of entropy, which is a way to measure the amount of randomness in a given set of data. Entropy is a term used in the fields of physics and mathematics to talk about how much a system is random or dirty. In the field of information theory, "impurity" means that a set of examples has parts that are different from each other. The concept of information gain pertains to the reduction of entropy. The process of information gain involves the calculation of the disparity between the entropy prior to a split and the average entropy following a split of a dataset, which is determined by the attribute values provided.

$$Info(D) = -\sum_{i=1}^{n} y_i log_2 y_i \ldots\ldots (1)$$

Here, for each given tuple D, y_i is the likelihood that it is a member of class C_i.

$$Info_A(D) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} \times Info(D_i) \ldots\ldots (2)$$

$$Gain(A) = Info(D) - Info_A(D) \dots\dots\dots (3)$$

The typical quantity of information required to determine a tuple's class label is denoted by Info(D). The i^{th} partition's weight is denoted by the expression |Di|/|D|. The anticipated information needed to place a tuple from D into one of the categories defined by A is denoted by $Info_A(D)$. At each node N(), the splitting attribute is determined by the attribute A with the biggest information gain, Gain(A).

4.1.3 Gain Rati

The measure of information gain exhibits bias towards attributes that possess a larger number of outcomes. This implies that it exhibits a preference for attributes that possess a significant quantity of unique values. One example to illustrate this concept is an attribute possessing a distinct identifier that contains no information due to complete partitioning. This approach maximises the amount of information obtained and results in the creation of

```
In [21]: from sklearn.naive_bayes import GaussianNB

         NaiveBayes = GaussianNB()

         NaiveBayes.fit(Xtrain,Ytrain)

         predicted_values = NaiveBayes.predict(Xtest)
         x = metrics.accuracy_score(Ytest, predicted_values)
         acc.append(x)
         model.append('Naive Bayes')
         print("Naive Bayes's Accuracy is: ", x)

         print(classification_report(Ytest,predicted_values))
```

Fig. 5. Naive Bayes Workflow code

partitions that are of no practical value. The Gain Ratio method addresses the potential problem of bias by standardising the information gain through the utilisation of Split Information.

$$SplitInfo_A(\mathrm{D}) = -\sum_{i=0}^{n} \frac{|D_i|}{|D|} \times log_2\left(\frac{|D_i|}{|D|}\right) .. (4)$$

Where |Di|/|D| represents the ith part's relative importance. There are n distinct values for the property A.

One definition of gain ratio is

$$GainRation(A) = \frac{Gain(A)}{SplitInfo_A(D)} \dots (5)$$

The characteristic with the highest gain ratio is picked to be the dividing factor.

4.1.4 Gini index

CART (Classification and Regression Tree) is a decision tree technique that use the Gini approach for determining branching.

$$Gini(D) = 1 - \sum_{i=1}^{n} y_i^2 \dots\dots (6)$$

If a pair in D has a certain probability (yi), then it must belong to class Ci.

The Gini Index assumes a binary distribution for all characteristics. Each partition's impurity may be added to the total via a weighted sum. When at-attribute values are either If A divides a data set D into two subsets, D1 and D2, then D's Gini index is:

$$Gini_A(D) = \frac{|D_i|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) .. (7)$$

If the property has discrete values, the subset with the lowest gini index is used to divide the data. For characteristics with continuous values, one approach is to find pairs of values that are close together and then choose the one with the smallest gini index.

$$\Delta Gini(A) = Gini(D) - Gini_A(A) \dots\dots (8)$$

The characteristic with the lowest Gini index is selected as the dividing factor.

4.2 Naive Bayes

The Naive Bayes algorithm is a way to classify things based on statistics that uses Bayes' Theorem. This algorithm is considered to be among the most straightforward forms of supervised learning. The Naive Bayes algorithm is an efficient and trustworthy method. When used on large datasets, the Naive Bayes method demonstrates impressive

accuracy and efficiency. The Naive Bayes algorithm makes its predictions on the premise that the presence or lack of additional characteristics does not affect the weight assigned to any one feature in determining a class. The desirability of a loan applicant is contingent upon various factors such as their income, prior loan and transaction history, age, and geographic location. Although these features may be interdependent, they are still regarded as independent features. The aforementioned assumption is regarded as naive due to its tendency to streamline calculations. The concept being referred to is known as class conditional independence.

$$p\left(\frac{h}{D}\right) = \frac{p\left(\frac{h}{D}\right)p(h)}{p(D)} \ldots\ldots (9)$$

P(h): How likely it is that hypothesis h is correct. (regardless of the data). Commonly known as the previous likelihood of hypothesis h, the above is a key statistic in many scientific fields.

P(D): probability based on the available information (regardless of the hypothesis). In the academic world, the aforementioned idea is more often known as the prior probability.

P (h/D): The probability of hypothesis h conditioned on the observed data. D. The term "posterior probability" is frequently used in academic discussions to refer to the aforementioned concept.

P (D/h): The conditional probability of data set d, given the assumption that hypothesis h holds true. The aforementioned is commonly referred to as posterior probability. There exist two distinct approaches, the first of which pertains to a singular feature, while the other approach pertains to multiple features.

4.2.1 First Approach

The Naive Bayes classifier figures out how likely something is to happen in a series of steps:

- Calculate the prior probability of the given class labels.
- The task at hand is to figure out how likely it is that each attribute will happen for each class.
- Put the given value into the Bayes formula to figure out the posterior probability.
- Find the class with the highest likelihood, given that the input belongs to the class with the highest likelihood.

4.2.2 Second Approach

The Naive Bayes classifier figures out how likely something is to happen in a series of steps:

- Determine the prior probability of a given set of class labels.
- The current task is to figure out the conditional probabilities for each attribute and each class.
- The task at hand involves the multiplication of same-class conditional probabilities.
- The prior probability and the probability found in step three should be multiplied together.
- Find the most likely class and assign it to the set of inputs you have at hand.

4.3 SVM

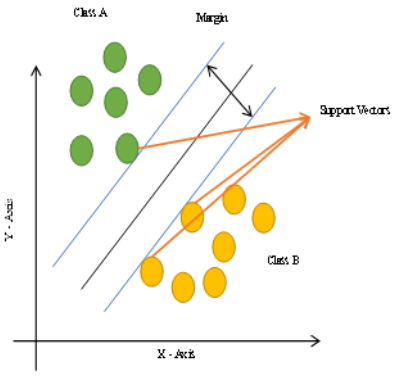

Fig. 6. SVM

The SVM is commonly recognized as a classification methodology; however, it is versatile enough to be utilized in both classification and regression scenarios. The software is capable of effectively managing a variety of continuous and categorical variables. SVM utilize the creation of a hyperplane within a multi-dimensional space to effectively distinguish and separate distinct classes. The SVM algorithm iteratively generates an optimal hyperplane that is utilized to minimize error. The fundamental concept underlying SVM is to identify an optimal hyperplane with maximum margin (MMH) that effectively separates the given dataset into distinct classes.

Support Vector Machine (SVM)

```
In [24]: from sklearn.svm import SVC

SVM = SVC(gamma='auto')

SVM.fit(Xtrain,Ytrain)

predicted_values = SVM.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

Fig. 7. SVM Workflow code

4.3.1 Support Vectors: Support vectors refer to the specific data points that exhibit the shortest distance to the

hyperplane. By figuring out the margins, these elements will be better able to see the line between the two groups. These are the aspects that are most important to consider when developing the classification.

4.3.2 Hyperplane: One definition of a decision plane describes it as a hyperplane if it divides a collection of objects into classes that are distinct from one another.

The main objective is to accurately classify the information that has been presented. The margin refers to the spatial gap between the two points that are in closest physical proximity to each other. The objective is to select a hyperplane that

4.3.3 Margin: A margin is the space that exists between two lines that represent the class points that are the nearest together. This is determined by measuring the distance in a perpendicular direction from the line to the support vectors or the locations that are nearest. If the gap between the classes is greater, then the margin is considered to be excellent; on the other hand, a gap that is smaller is considered to be a poor margin

maximises the distance between the support vectors based on the given information. The following are the stages undertaken by Support Vector Machines (SVM) during the search for the hyperplane with the highest residual.

Logistic Regression

```
In [26]: from sklearn.linear_model import LogisticRegression

         LogReg = LogisticRegression(random_state=2)

         LogReg.fit(Xtrain,Ytrain)

         predicted_values = LogReg.predict(Xtest)

         x = metrics.accuracy_score(Ytest, predicted_values)
         acc.append(x)
         model.append('Logistic Regression')
         print("Logistic Regression's Accuracy is: ", x)

         print(classification_report(Ytest,predicted_values))
```

Fig. 8. Logistic Regression Workflow code

1. Produce hyperplanes that separate the classes in the most effective manner possible. The picture on the left-hand side depicts three hyperplanes coloured black, blue, and orange. In this example, the blue and orange have a greater categorization mistake than the black does, which accurately differentiates between the two classifications.
2. Select the hyperplane that exhibits the maximum degree of separation from the nearest reference points, as illustrated in the image located on the right-hand side.

Dealing with non-linear and inseparable planes: Dealing with non-linear and indivisible planes: As shown in the picture below, linear hyperplanes are not always the best tool to use when trying to handle certain situations. (left-hand side). When this occurs, the SVM employs a kernel technique to convert the input space to a space with greater dimensions, as shown on the right. The data values are displayed on the x-axis and the z-axis (z is the squared total of both x and y, which can be written as z = x2 + y2). The use of linear separation will now make it simple for you to separate these elements.

4.4 Logistic Regression

The statistical technique known as logistic regression is used to make predictions regarding binary classifications. The essence of the consequence or objective variable is that

Random Forest

```
In [29]: from sklearn.ensemble import RandomForestClassifier

         RF = RandomForestClassifier(n_estimators=20, random_state=0)
         RF.fit(Xtrain,Ytrain)

         predicted_values = RF.predict(Xtest)

         x = metrics.accuracy_score(Ytest, predicted_values)
         acc.append(x)
         model.append('RF')
         print("RF's Accuracy is: ", x)

         print(classification_report(Ytest,predicted_values))
```

Fig. 9. Random Forest Workflow code

it is indeterminate. When we talk about something being dichotomous, we imply that there are only two potential classifications. For instance, it may be applied to issues concerning the diagnosis of malignancy. It determines the likelihood that a certain incident will take place. In this particular application of linear regression, the variable of

interest is one that is of a categorical character. As the dependent variable, it makes use of a log of the chances. A logit function is utilised in the process of logistic regression, which forecasts the likelihood of recurrence of a binary event. The Equation for Linear Regression Is:

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n \quad \ldots\ldots (10)$$

Here, y is dependent variable and x_1, x_2 ... and x_n are explanatory variables.

Sigmoid Function:

$$f(x) = \frac{1}{1 + e^{-(x)}} \quad \ldots . (11)$$

Linear regression using a sigmoid function:

$$f(x) = \frac{1}{1 + e^{-(a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n)}} \ldots (12)$$

Logistic Regression Characteristics:

In logistic regression, the dependent variable is modelled after a Bernoulli distribution.

Maximum likelihood is used for the estimation process.

Concordance, rather than R squre, is used to measure model fitness. KS-Statistics.

4.5 Random Forest

Specifically, it is a technique known as a decision tree ensemble, and it is built on the divide-and-conquer approach. This technique was developed using a dataset that was arbitrarily divided. This collection of decision tree classifications also goes by the moniker the forest in some contexts. The individual decision trees are generated by applying an attribute selection indicator to each attribute. Some examples of such indicators include information gain, gain ratios, and the Gini index. Every single branch is dependent on its own individual random selection. Each branch involved in a categorization problem has one vote, and the solution will eventually be chosen from the category that receives the most votes. In the case of a regression, the eventual result is determined to be the same as the mean of all of the tree's results. It becomes clear when compared to other non-linear categorization techniques that not only is it simpler, but that it is also more successful**.**

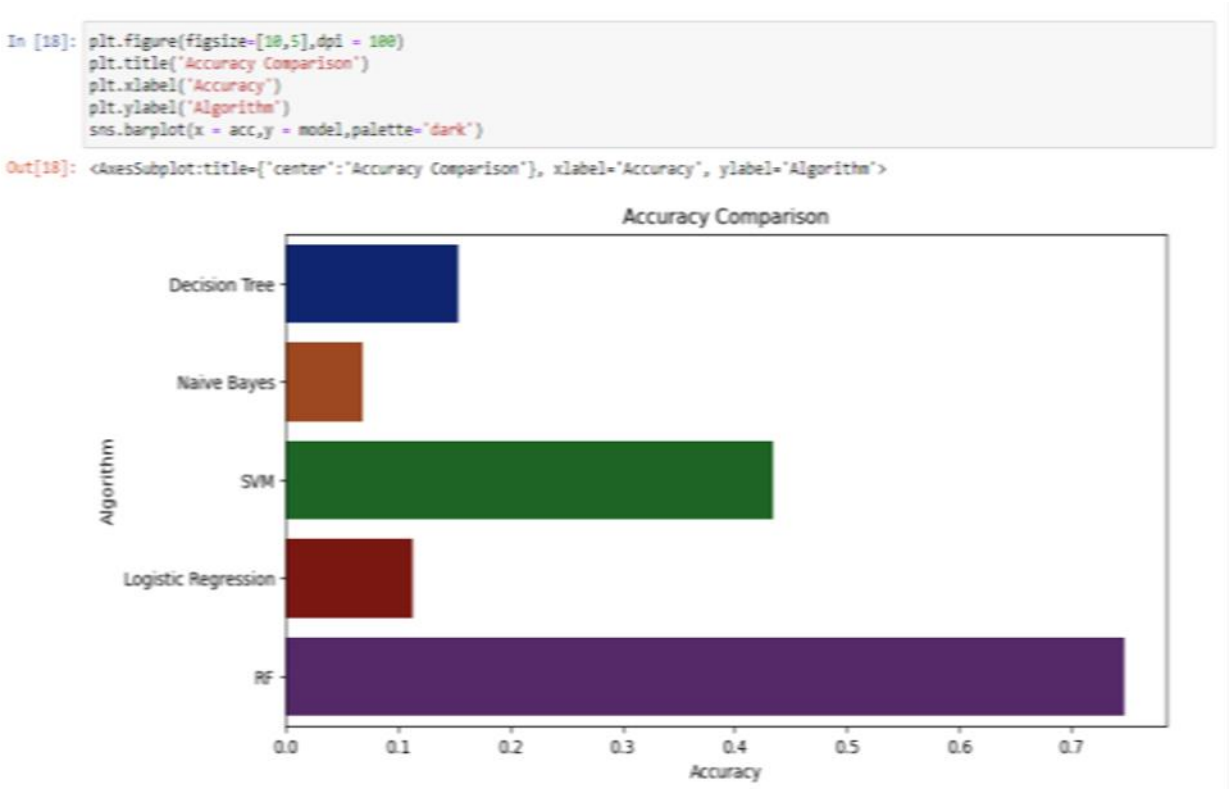

Fig. 10. Accuracy

It works in four steps:

1. Using the information provided, select some examples at random.

2. Construct a decision tree for each of the samples, and then obtain an outcome of projection from each of the decision trees.

3. Take an action to find out whether your forecasts came true.

5. Experiment

The Jupyter notebook serves as the host for our experimental code, and our physical system configuration includes an Intel Core i7-10750H CPU running at 2.60 GHz,

16 GB of RAM, and 512 GB of solid-state storage.

Prerequisites for the software include:

5.1 Jupyter Notebook: An open-source web application called Jupyter Notebook lets users make and share documents that include live code, mathematical expressions, visualizations, and written text. The Jupyter Notebook is a web-based software application. Python programming can be facilitated through the pre-installed IPython kernel in Jupyter. It is noteworthy that there are more than 100 alternative kernels available for exploitation. The Jupyter Notebook integrates the following three elements:

5.1.1. The notebook web application: A web-based application that allows for the writing and execution of code in a collaborative manner as well as the production of notebook documents.

5.1.2. Kernels: Users' code is executed in a specific language using separate processes that are initiated by the notebook web application. The result from these processes is then returned to the notebook web application. A number of other tasks, including calculations for interactive interfaces, tab completion, and introspection, are managed by the kernel.

5.1.3. Notebook documents: The aforementioned refers to self-sufficient documents that encompass a comprehensive representation of the content available in the online notebook application. The content in question may comprise various elements such as inputs and outputs of computations, descriptive text, mathematical equations, images, and multimedia depictions of objects. Each document within the notebook possesses its own unique engine.

5.2. Python:

The aforementioned is a programming language that is object-oriented and high-level, featuring dynamic semantics that are built-in, and is mostly used to make websites and apps for mobile devices. It is a very good choice for rapid application development because it gives you options for dynamic encoding and dynamic linking. Python is a relatively simple programming language. Because of this, it is not hard to learn, even though it has a different grammar that puts an emphasis on readability. Python code is much simpler to understand and interpret for developers than code written in other languages. Because Python allows for the use of both modules and packages, it is possible for programmers to be constructed in a flexible fashion and for code to be repurposed across a number of different projects.

5.3 Visual Studio Code:

It is a code editor that can be used on macOS, Linux, or Windows. It was made by Microsoft. Some of the features are help with troubleshooting, grammar highlighting, intelligent code completion, code excerpts, code restructuring, and an integrated Git. Users have the ability to alter the look and feel of the application, customize their

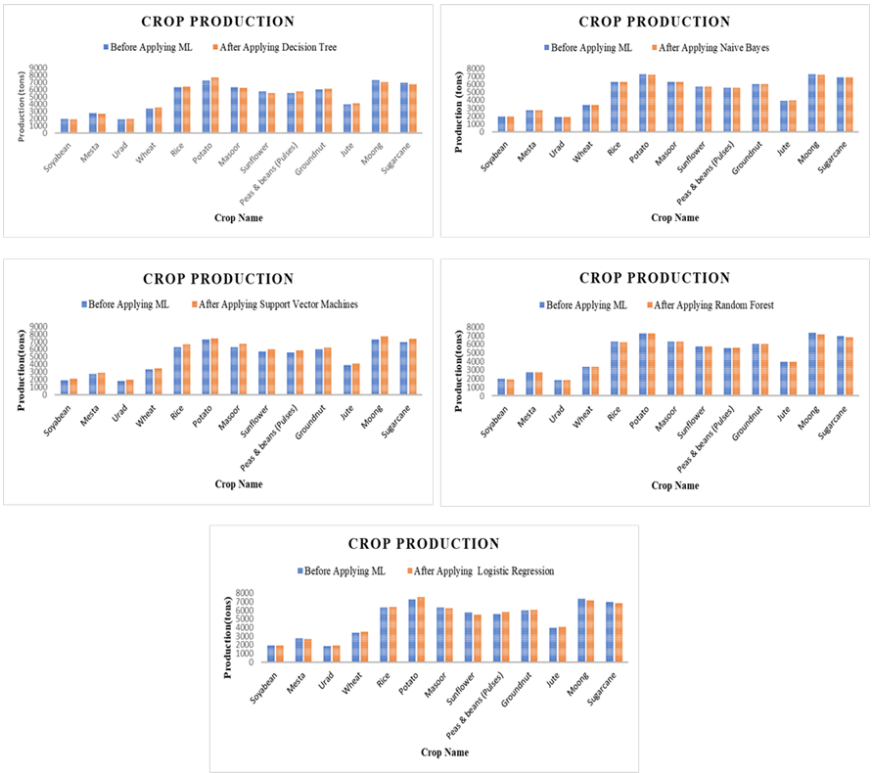

Fig. 11. Crop Production of Different ML

computer functions and preferences, and install extensions that extend the application's capabilities.

5.4 Flask: Flask is an efficient framework for developing WSGI online applications. It is intended to make it simple and fast to get started while also having the capacity to build up to support more complicated applications. It started out as a straightforward container around Werkzeug and Jinja, but it has since grown to become one of the most widely used frameworks for Python online applications. It does not impose any prerequisites or project structure requirements, but it does offer recommendations. It is up to the developer to decide which tools and frameworks they will make use of in their project. There are many enhancements that have been developed and supplied by the community, which makes it simple to add new functionality.

6. Results

In order to accomplish the goals of this study, we implemented a variety of well-known algorithms, including decision trees, naïve bayes, SVM, logistic regression, and random forests. Every single one of the algorithms is founded on the concept of supervised learning. In figure 11, the crop production of five ML algorithms is displayed. Rice, potato, and wheat production are measured in thousands of tones, while wheat production is measured in tones. It will increase production for each technique, while random forest and SVM produce more effectively compared to other models. Following the completion of the dataset training, we will determine the accuracy of this algorithm and then compare it to the accuracy of the other algorithms. In this case, we discover that the Random Forest algorithm provides the highest level of precision for our dataset. Following the completion of the training dataset, the Random Forest algorithm was utilized to make a prediction of the harvest based on the value of weather data including temperature, rainfall, humidity, and sun hours.

7. Conclusion and Future Works

The most advantageous aspect of applying ML in agriculture is that it will not cause human farmers' employment to be eliminated; rather enhance the processes that farmers currently use. This method assists the farmer in selecting the appropriate crop by supplying insights that are not typically kept track of by conventional farmers; as a result, the likelihood of agricultural failure is reduced, and overall productivity is increased. Using five different ML-based models, we were able to make agricultural production forecasts based on the meteorological data. The proposed work has only been tested on two district datasets. In the future, we will try to collect data worldwide for training and testing. The work that will be done in the future will centre on supplying the succession of products that should be produced depending on the circumstances of the land and the weather, as well as on regularly updating the databases in order to make accurate projections. The goal of the work that will be done in the future is to develop a completely automatic system that will do the same thing and capture reliable meteorological data using gadgets connected to the internet of things.

Acknowledgements

The department of computer science at The University of Burdwan, located in Burdwan, West Bengal, India, provided technical support for this research article. Our sincere gratitude goes out to Dr. Sunil Karforma, dean of science at the University of Burdwan, for his assistance in guiding us.

Author contributions

The conception and design of the study were collective efforts from all of the authors. Saikat Banerjee was in charge of the material preparation, data collection, and analysis while Abhoy Chand Mondal was present to provide direction and feedback. Both of the authors offered their thoughts and observations on earlier draughts of the manuscript. The final manuscript was reviewed and approved by both of the authors.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Thomas van Klompenburga, Ayalew Kassahuna, Cagatay Catalb “Crop yield prediction using machine learning: A systematic literature review” Computers and Electronics in Agriculture 177, 2020 Elsevier, DOI: 10.1016/j.compag.2020.105709.

[2] Ansarifar, J., Wang, L. & Archontoulis, S.V. An interaction regression model for crop yield prediction. Sci Rep 11, 17754 (2021). https://doi.org/10.1038/s41598-021-97221-7.

[3] Pallavi Kamath, Pallavi Patil, Shrilatha S, Sushma, Sowmya S(2021), “Crop yield forecasting using data mining”, Global Transitions Proceedings, Volume 2, Issue 2, Pages 402-407

[4] Kiran Moraye, Aruna Pavate, Suyog Nikam and Smit Thakkar(2021), “Crop Yield Prediction Using Random Forest Algorithm for Major Cities in Maharashtra State”, International Journal of Innovative Research in Computer Science & Technology (IJIRCST), ISSN: 2347-5552, Volume-9, Issue-2.

[5] Ms. Fathima, Ms. Sowmya K, Ms. Sunita Barker, Dr. Sanjeev Kulkarni(2020), “Analysis of Crop yield Prediction using Data Mining Technique” International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 05.

[6] A. Gonzalez-Sanchez, J. Frausto-Solis, and W. Ojeda-Bustamante(2014): "Predictive ability of machine learning methods for massive crop yield prediction", Spanish Journal of Agricultural Research, vol. 12, no. 2, pp. 313–328.

[7] D. K, R. M, S. V, P. N, and I. A. Jayaraj(2021), "Meta-Learning Based Adaptive Crop Price Prediction for Agriculture Application," in 2021 IEEE, 5th International Conference on Electronics, Communication, and Aerospace Technology (ICECA), pp. 396-402, DOI: 10.1109/ICECA52323.2021.9675891.

[8] E. Khosla, R. Dharavath, and R. Priya, "Crop yield prediction using aggregated rainfall-based modular artificial neural networks and support vector regression," Environment, Development and Sustainability, pp. 1–22, 2019.

[9] T. U. Rehman, M. S. Mahmud, Y. K. Chang, J. Jin, and J. Shin, "Current and future applications of statistical machine learning algorithms for agricultural machine vision systems," Computers and electronics in agriculture, vol. 156, pp. 585–605, 2019

[10] Shanthi Selvaraj(&), Poonkodi Palanisamy, Summia Parveen, and Monisha (2016): "Autism Spectrum Disorder Prediction Using Machine Learning Algorithms", Springer Nature Switzerland AG 2020 S. Smys et al. (Eds.): ICCVBIC 2019, AISC 1108, pp. 496–503.

[11] Snehal S.Dahikar, Dr.Sandeep V.Rode(2014): "Agricultural Crop Yield Prediction Using Artificial Neural Network Approach", International journal of innovative and research in electrical, instrumentation and control engineering, volume 2,Issue 2.

[12] N.Gandhi, L.J. Armstrong and O. Petkar(2016): "Rice Crop Yield Prediction in India using Artificial Neural Network", International Conference on 2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), Chennai, India scheduled on 15th and 16th July.

[13] Takeshi Yoshida Noriyuki Murakami and Hiroyuki Tauiji.(2017): "Hybrid Machine Learning Approach to Automatic Plant Phenotyping For Smart Agriculture". 978-1- 5090-5888-4/16/$31.00 @IEEE.

[14] S.Veenadhari Dr. Bharat Misra Dr. CD Singh(2014):" MLapproach for forecasting crop yield based on climatic parameters", 2014 International Conference on Computer Communication and Informatics (ICCCI -2014), Jan. 03 – 05, Coimbatore, INDIA.

[15] Gulati P and Jha S K (2020): "Efficient crop yield prediction in India using machine learning techniques", International Journal of Engineering Research & Technology (IJERT) ENCADEMS – vol 8 Issue 10.

[16] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi(2014): "Identification and classification of fungal disease affected on agriculture/horticulture crops using image processing techniques", IEEE International Conference on the Computational Intelligence and Computing Research.

[17] N. R. Prasad, N. R. Patel, and A. Danodia (2021): "Crop yield prediction in cotton for regional level using random forest approach", Spat. Inf. Res., vol. 29, no. 2, pp. 195– 206, doi: 10.1007/s41324-020-00346-6.

[18] S. B. Jadhav and S. B. Patil(2015): "Grading of Soybean Leaf Disease Based on Segmented Image Using K-means Clustering", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), vol. 4, no. 6.

[19] Rushika Ghadge, Juilee Kulkarni, Pooja More, Sachee Nene, Priya R L(2018): "Prediction of crop yield using machine learning", International Research Journal of Engineering and Technology.

[20] Nigam A, Garg S, Agrawal A, Agrawal P (2019): "Crop yield prediction using machine learning algorithms", Fifth international conference on image information processing (ICIIP). IEEE, pp 125–130.

[21] John William Orillo, Gideon Joseph Emperador, Mark Geocel Gasgonia, Marifel Parpan, and Jessica Yang(2014): "Rice plant nitrogen level assessment through image processing using artificial neural network", IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), pp. 1-6.

[22] Kalimuthu M, Vaishnavi P and Kishore M (2020): "Crop prediction using machine learning", 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) pp 926-32 doi: 10.1109/ICSSIT48917.2020.9214190.

[23] Mondal, Akash, and Saikat Banerjee. "Effective Crop Prediction Using Deep Learning." 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). IEEE, 2021.

[24] S Banerjee, S Chakraborty, AC Mondal, Machine Learning Based Crop Prediction on Region Wise Weather Data, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 1.

[25] Alexandre Barbosa, Naira Hovakimyan, Nicolas F. Martin(2020): "Risk_averse optimization of crop

inputs using a deep ensemble of convolutional neural networks", Available online 1 October 2020. https://doi.org/10.1016/j.compag.2020.105785.

[26] S.Veenadhari, Dr Bharat Misra, Dr CD Singh.(2019): "Machine learning approach for forecasting crop yield based on climatic parameters", 978-1-4799-2352-6/14/$31.00 ©IEEE.

[27] S Banerjee and A. C. Mondal. "An Intelligent Approach to Reducing Plant Disease and Enhancing Productivity Using Machine Learning", International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 3, Apr. 2023, pp. 250-62,

[28] Khare, S. ., & Badholia, A. . (2023). BLA2C2: Design of a Novel Blockchain-based Light-Weight Authentication & Access Control Layer for Cloud Deployments. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 283–294. https://doi.org/10.17762/ijritcc.v11i3.6359

[29] Dr. Bhushan Bandre. (2013). Design and Analysis of Low Power Energy Efficient Braun Multiplier. International Journal of New Practices in Management and Engineering, 2(01), 08 - 16. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/12