GMR
The Original

A RELIABLE MACHINE LEARNING FRAMEWORK FOR CROP RECOMMENDATION IN PRECISION AGRICULTURE

[1]Vemula Jasmine Sowmya, [2]Tarini Prasad Panigrahy ,[3] Kaliprasanna Swain

[1]Research scholar, Department of Computer Science and Engineering, GITA Autonomous College
(affiliated to Biju Patnaik University of Technology (BPUT), Badaraghunathpur, Madanpur, Near Janla, Bhubaneswar, Odisha.
[2]Department of Computer Science and Engineering, GITA Autonomous College, Badaraghunathpur, Madanpur, Near Janla, Bhubaneswar, Odisha.
[3]Department of ETC, Trident Academy of Technology F-2, Chandaka Industrial Estate, In front of Infocity, Infocity , Chandrasekharpur, Bhubaneshwar, Odisha - 751024.
Emails: [1] vemulajasmine@gmail.com, [2]tpanigrahy@gmail.com, [3]kaleep.swain@gmail.com

ABSTRACT
Agriculture is the backbone of the Indian economy, providing livelihoods for a significant portion of the population. Selecting the most suitable crop based on soil and environmental conditions is essential for improving agricultural productivity and ensuring sustainable farming practices. Recent advancements in Machine Learning (ML) have enabled the development of intelligent decision support systems that assist farmers in making informed crop selection decisions. This study presents a machine learning-based crop recommendation system that predicts the most suitable crop using key soil and climatic parameters. The proposed model considers factors such as nitrogen (N), phosphorus (P), potassium (K), soil pH, temperature, humidity, and rainfall to generate accurate crop recommendations. Several supervised machine learning algorithms, including Random Forest (RF) and Extreme Gradient Boosting (XGBoost), are employed to analyze agricultural data and identify the most appropriate crop for specific field conditions. By integrating data-driven
insights with precision agriculture, the proposed system helps farmers improve crop yield, optimize resource utilization, and support sustainable agricultural practices.

KEYWORDS— Crop Recommendation, Precision Agriculture, Machine Learning, Supervised Learning, Random Forest, XGBoost, Soil Nutrients (NPK), Soil pH, Sustainable Agriculture.

I. INTRODUCTION

Agriculture is crucial to India's economy and existence. It's a crucial position. It also affects our daily lives. [1]. Farmers commit suicide in the majority of cases because they are unable to repay bank debts taken for farming purposes, resulting in a loss of production[2].
The goal of this study is to select the best crop depending on input characteristics such as nitrogen (N), phosphorus (P), potassium (K), soil PH, humidity, temperature, and rainfall.
This paper uses various supervised machine learning approaches to predict the accuracy of future production of eleven different crops in India, including all major crops and various soil characteristics including ph of soil, humidity of soil, rainfall(in mm) of soil and NPK values of the soil.
Various Supervised ML Algorithms like SVM, Random Forest, etc.. Were all being used in this suggested system? As a result, we may use recommendation system technology to identify crops that are suited for growing as well as how to maximize the crop yield and profit. We can help farmers become more technically sound by assisting them in spreading awareness among their fellow farmers about the best crop to plant, and we can also help them become financially secure by preventing crop loss every year by suggesting the best crop according to their soil type.
Overview of remaining sections is, Section II delves into previous research. Section III one describes the background technique used to build the prediction model. Section IV and V describes the experimental results and observations, whereas Sections VI and VII covers the conclusion and future work.

II. LITERATURE SURVEY

Farmers encounter problems due to incorrect crop selection in the type of soil available to them, according to S. Pudumalar et al. [3]. This is due to a lack of relevant expertise. They devised a suggestion system that can assist farmers in selecting the best crop for their soil. After a series of experiments, they concluded that the system built can anticipate acceptable results to a reasonable and decent level with an accuracy of 88 percent using techniques such as Random Forest, Naive Bayes, CHAID model, and K-nearest neighbor. Despite the fact that they tested the model for a specific condition, the model can produce a more accurate recommendation with virtually any changes to the information.
R. Kumar et al. described a technique that uses a farmer's location to anticipate the crop that would be suited in that area by identifying his or her position and then working with various climatic and economic variables at the sub-district level [4].Its objective is to gather data about that location, such as temperature, seasonal crops, and crop growing season, and then to identify similar upazilas based on the aforementioned factors and the seasonal

crop to be grown, and then to select the top-k crops from among them. he model's front end was created with XML, and the model's back end was created with JAVA and Android Studio. The created model forecasts a satisfactory crop.
[6] Suresh.G et al. "Digital Farming: " A sample datasetfor location data and a sample dataset for crop data were used in this investigation. This proposed approach advised specific crops based on their nutrient (N, P, K, and PH) values, as well as identifying available nutrient values and required fertilizer levels for specific crops such as rice, maize, black gramme, carrot, and radish.
[5] D. Anantha et al. suggested a method that relied on three elements: Properties of Soil, Types of Soil, and yield of crop data gathering. Based on these three parameters the crop is being recommended to farmers. RF, KNN, and Naive Bayes are just a few of the ML algorithms employed in this system. We can predict specific crop values, state and district values, by using the proposed approach. As a result, our proposed effort would assist farmers in sowing the appropriate seed based on soil conditions in order to boost national output.

III.PROPOSED WORK

The proposed work includes a process that has been divided into several steps as shown in fig1. The following are the three phases:

1) Datasets collection
2) Extraction of Features
3) Used a number of machine learning algorithms

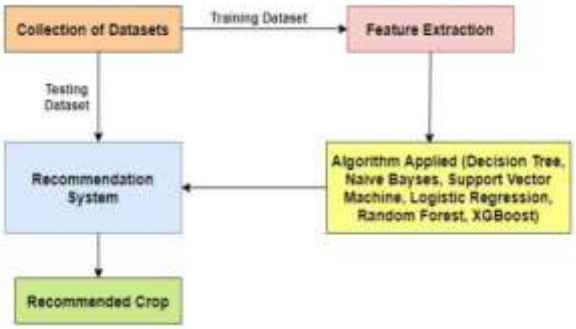

Fig 1: Block diagram of proposed system

(1) Dataset Collection:
The data used in this project is made by augmenting and combining various publicly available datasets in India like weather, soil, etc.. The data is relatively simple with very few but useful features unlike the complicated features affecting the yield of the crop. The dataset contains 2200 rows of data and 8 columns that are usually the features describing the soil like Nitrogen-Phosphorus-Potassium(NPK), soils humidity, rainfall, temperature, and ph that describe the nature of the soil.For training and testing, Dataset has been partitioned into two different sizes. Remaining 20% of dataset is used to evaluate each ML model and calculate its metrics.

(2) Extraction of Features:
The Extraction of Features has a huge effect on machine learning. Several researchers have found that certain features are helpful in training machine learning-based classifiers. This is why, in our solution, we use feature extraction.. In supervised learning, we will use extraction of features, which is the process of selecting, manipulating, and changing raw data into features. We use feature extraction for the following reasons:
• To remove imputation
• Handling outliers
• To achieve Normalization
• Null Value Handling
• To remove invalid data

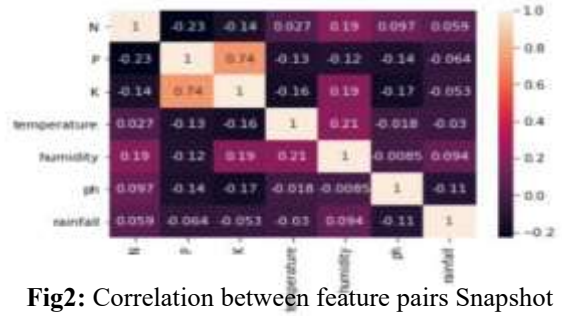

Fig2: Correlation between feature pairs Snapshot

```
Decision Tree --> 0.9
Naive Bayes --> 0.990909090909091
SVM --> 0.10681818181818181
Logistic Regression --> 0.9522727272727273
RF --> 0.990909090909091
XGBoost --> 0.9931818181818182
```

(3) Methodology:
In this proposed system, applied various supervised Machine Learning algorithms to select the best accurate algorithm that predicts the most suitable crop to the soil conditions.

A. Decision Tree:
The sole objective of Decision Tree is to construct a machine learning model that guesses the class of a given instance by learning basic decision rules from feature values. First, we will calculate the entropy. Is also known as a measure of uncertainty. Then for each attribute A, we calculate information gain. The root node will be chosen based on the attribute with the highest value for knowledge gain, and the process will repeat.. The formulas for entropy and Information gain are:

$E(S) = -[p\log(p) + (1-p)\log(1-p)]$ $Gain(S, A) = E(S) - v\ p_v E(S_v)$

Applied this algorithm into our model as follows:
(i) Importing library DecisionTreeClassifier from sklearn.tree Class.
(ii) Now we create a DecisionTree Classifier object.

(iii) In the end we fit our data.

```
# Decision Tree
from sklearn.tree import DecisionTreeClassifier
DecisionTree = DecisionTreeClassifier(criterion
="entropy", random_state=2, max_depth=5)
DecisionTree.fit(Xtrain,Ytrain)
```

B. Naive Bayes(NB):
A set of classification algorithms based on the Bayes theorem is known as a Naive Bayes classifier. It's necessary to have more than one. It's a set of algorithms that all follow the same basic principle: Each categorization feature pair is distinct from the others.. Bayes theorem says that for two events Ea, Eb:
$P(Ea|Eb) = P(Eb|Ea) * P(Ea)/ P(Eb)$

We can say, using this approach, that: $P\ r(class|attributes) = P\ r(attributes|class) * P\ r(class)/P\ r(attributes)$ Applied this algorithm into our model as follows:

(i) Importing library from sklearn.naive_bayes Class.
(ii) Now we create a Naïve Bayes Classifier object.
(iii) In the end we fit our data.

```
# Naive Bayses (NB)
from sklearn.naive_bayes import GaussianNB
NaiveBayes = GaussianNB()
NaiveBayes.fit(Xtrain, Ytrain)
```

C. Support Vector Machine(SVM):

A SVM Linear classifier is built to fit the data you supply and provide a hyperplane that fits well and classify your data into different classes. Following that, you may input some attributes to your classifier to check what the projected class is once you've obtained the hyperplane. Support Vectors are the points which can be considered as edge cases. They are very close to the hyperplane. The two support vectors corresponding to either classes benign and malicious respectively are equidistant to the hyperplane with maximum margin possible.

Applied this algorithm into our model as follows:

(i) Importing library SVC from sklearn.svm Class. (ii) Now we create an SVM classification object.
(iii) At last we fit our data.

```
# Support Vector Machine(SVM)
from sklearn.svm import SVC
SVM = SVC(gamma='auto')
SVM.fit(Xtrain, Ytrain)
```

D. Logistic Regression:

```
Decision Tree --> 0.9
Naive Bayes --> 0.990909090909091
SVM --> 0.10681818181818181
Logistic Regression --> 0.9522727272727273
RF --> 0.990909090909091
XGBoost --> 0.9931818181818182
```

It's a machine learning algorithm for classifying problems.. Here, we will have a hypothesis function which will tell us the probability that a sample belongs to a particular class. The sigmoid function is employed as the hypothesis function in logistic regression.

$h_\theta(x) = \sigma(x^T\theta)$

$\sigma(x^T\theta) = 1/1 + e^{-xT_\theta}$

This classifier, in an ideal world, would be able to predict the outcome flawlessly. We must first find a function that minimizes J().. We do that using gradient descent algorithm

$J(\theta) = -1/m\ m\ i=1\ [(1 - y^i)\log(1 - p^i) + y^i \log(p^i)]$ where probability denoted by p^i and labeled class by y^i

Applied this algorithm into our model as follows:

(i) Importing library LogisticRegression from sklearn.linear Class
(ii) Now we create LogReg classifier object
(iii) In the last we fit our data

```
from sklearn.linear_model import LogisticRegre
ssion
LogReg = LogisticRegression(random_state=2)
LogReg.fit(Xtrain, Ytrain)
```

E. Random Forest(RF):

The random forest is a supervised learning method. It generates a "forest" from a set of numerous trees. These trees are often trained using an approach called bagging approach. The key premise of this method is that merging multiple learning models improves overall output. Random forest, in simple terms, is a method of merging numerous decision trees to provide a more accurate and dependable prediction. Random Forest is a machine learning classifier that uses a variety of decision trees on different subsets of input datasets to improve model performance. Our model's performance will improve as we add additional decision trees. Applied this algorithm into our model as follows:

(i) Importing library RandomForestClassifier from sklearn.ensemble Class
(ii) Now we create RF classifier object
(iii) In the last we fit our data

```
# Random Forest
from sklearn.ensemble import RandomForestCl
assifier
RF = RandomForestClassifier(n_estimators=20,
random_state=0)
RF.fit(Xtrain,Ytrain)
```

F. XGBoost:

XGBoost is a faster and more performant type of the gradient boosting method. XGBoost is known to show comparatively lesser results when compared to other algorithms of the same category. Nowadays it is being used in solving real world problems related to data science more quickly and accurately. It is an open-Source library.
Applied this algorithm into our model as follows:
(i) Importing library xgboost
(ii) Now we create XB classifier object
(iii) In the last we fit our data

```
# XGBoost
import xgboost as xgb
XB = xgb.XGBClassifier()
XB.fit(Xtrain, Ytrain)
```

IV. EXPERIMENTAL RESULT

Here we compare all the above used algorithms on the dataset and compare the accuracy and select the algorithm that produces best accurate output for the given input.

The comparison of the above mentioned algorithms were shown in bar chart below.

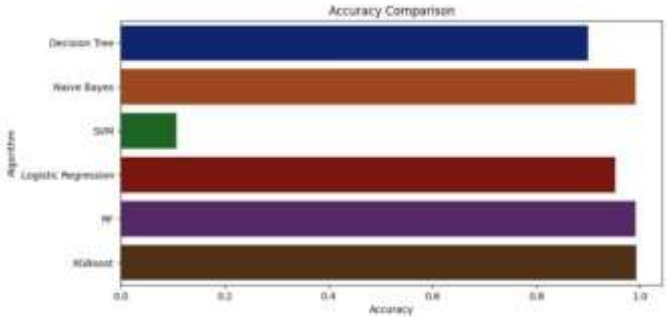

Fig 4: Accuracy Comparison

The RF algorithm outweighs all of the remaining ones in terms of accuracy. So the model is built on Random Forest and saved using a pickle module and predictions are done based on this model.

Recommending crops:

After selecting an accurate model, the findings were tested in a Jupyter notebook. Later, implemented this model in an application using the Flask platform.

```
# Making Prediction

data = np.array([[104,18,30,23.603016,60.3,6.7,140.91]])
prediction = RF.predict(data)
print(prediction)
```

```
['coffee']
```

Fig 5: Making Prediction Snapshot

```
data = np.array([[83,45,60,28,70.3,7.0,150.9]])
prediction = RF.predict(data)
print(prediction)
```

```
['jute']
```

Fig 6: Prediction Snapshot

Here, after giving the input array containing NPK values, Soil temperature, humidity, rainfall and ph values the model is predicting the best crop for that particular soil characteristics to yield maximum yield and profit.

V. RESULTS AND OBSERVATIONS

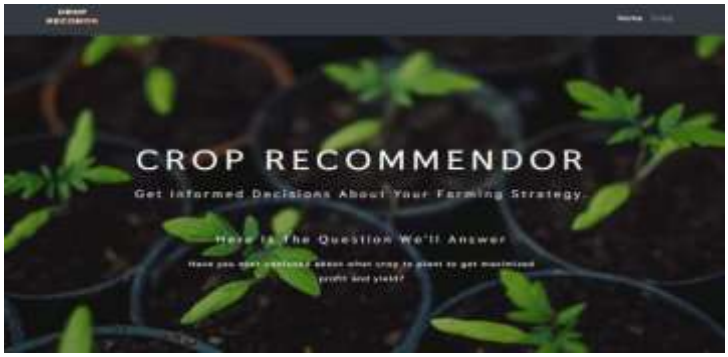

Fig 7: Web application Interface

After clicking the Crop link on the header of the application It takes us to the crop prediction page where the user gets to input as shown in the figure 8.

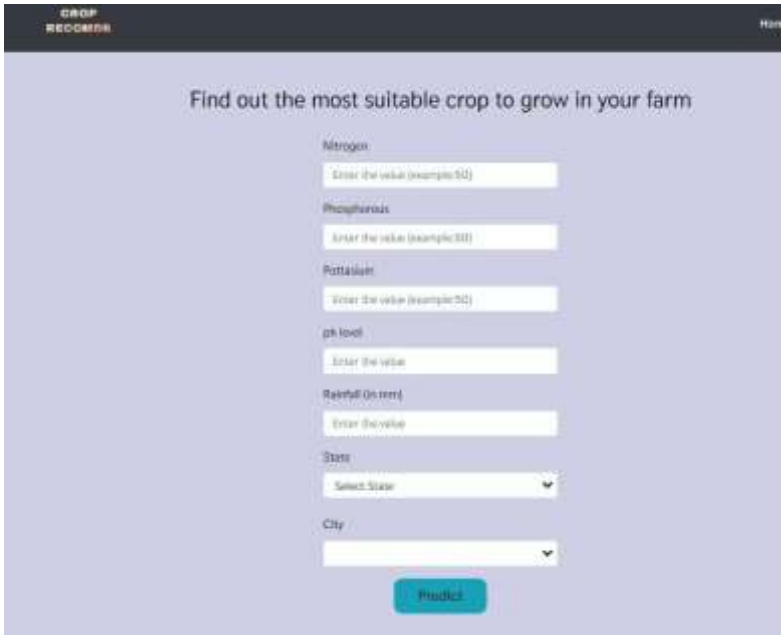

Fig 8: User Input Interface Snapshot

The user can enter his or her input soil data on this page.
The web app was built using Flask.

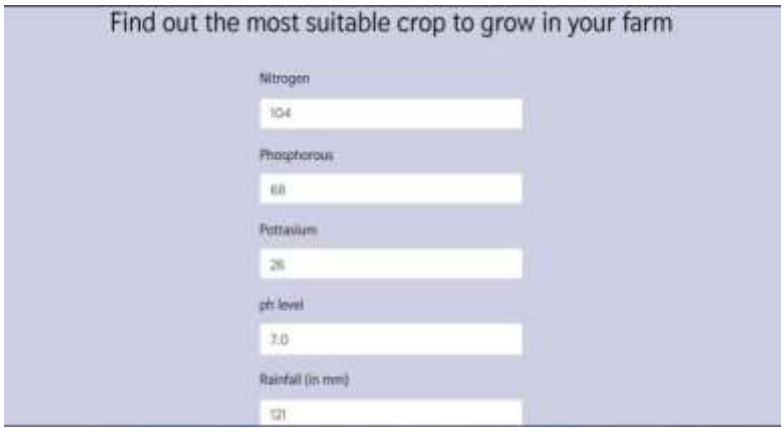

Fig 9: User entering inputs snapshot

After entering the inputs and clicking on the predict button, itredirects to the page of the predicted or recommended crop result as shown below.

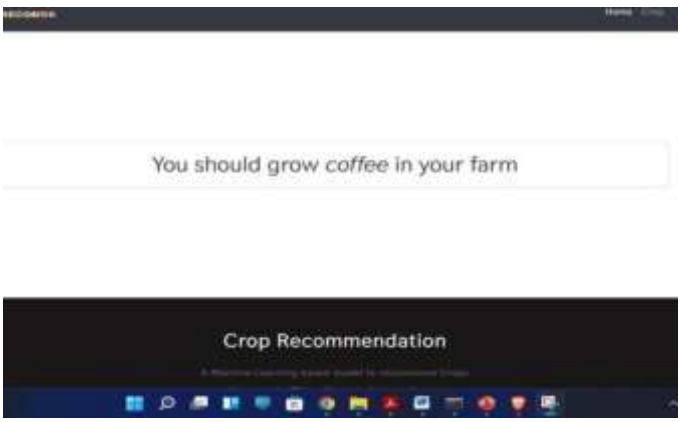

Fig 10: Result page Snapshot.

VI. CONCLUSION

Farmers today have numerous challenges, and they lack sufficient knowledge of which crops to grow and cultivate. This proposed approach assists farmers in determining the best crop to plant. The proposed approach uses data mining techniques to anticipate crops based on soil test findings. Agriculture departments can use this approach to predict the right crop at the right time. Farmers and the agricultural field will benefit from this type of automation. The developed system accomplished the following goals: it minimized manual effort, it can store vast volumes of data, and it has a seamless workflow.

VII. FUTURE WORK

The following features could be added to the system to make it even better:

1. Adding more attributes to the dataset. By doing so, it enriches the dataset.
2. We need to create a model that can distinguish between healthy and damaged crop leaves, as well as anticipate which disease is present if the crop has one.
3. To create an easy-to-use mobile app.

REFERENCES

[1] Kumar, Y. Jeevan Nagendra, V. Spandana, V. S. Vaishnavi, K. Neha, and V. G. R. R. Devi are among the authors. "In the Agriculture Sector, a Supervised Machine Learning Approach for Crop Yield Prediction." 736-741 in 2020 International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020.

[2]Garanayak, Mamata, Goutam Sahu, Sachi Nandan Mohanty, and Alok Kumar Jagadev are the authors of this article "Agricultural Recommendation System for Crops Using Different Machine Learning Regression Methods". International Journal of Agricultural and Environmental Information Systems (IJAEIS) 12, no. 1 (2021):1-20.

[3] S. Pudumalar, E. Ramanujam, R. Harine Rajashree, C. Kavya, T.Kiruthika, J. Nisha, “Crop Recommendation System for Precision Agriculture”, IEEE International Conference on Advanced Computing,Chennai, India, 2017, pp. 32-36.

[4] R. Kumar, M. P. Singh, P. Kumar, J. P. Singh, “Crop Selection Method to Maximize Crop Yield rate using Machine Learning Technique”, IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, Chennai, India, 2015, pp. 138-145.

[5] . Reddy, D. Anantha, Bhagyashri Dadore, and Aarti Watekar. "Crop recommendation system to maximize crop yield in ramtek region using machine learning." International Journal of Scientific Research in Science and Technology 6, no. 1 (2019): 485-489.

[6] . Suresh, G., A. Senthil Kumar, S. Lekashri, and R. Manikandan. "Efficient Crop Yield Recommendation System Using Machine Learning For Digital Farming." International Journal of Modern Agriculture 10, no. 1 (2021): 906-914.