Crop prediction using machine learning

Madhuri Shripathi Rao[1], Arushi Singh[1], N.V. Subba Reddy[1] and Dinesh U Acharya[1]

[1]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal, 576104, Udupi district Karnataka, India

Abstract: For most developing countries, agriculture is their primary source of revenue. Modern agriculture is a constantly growing approach for agricultural advances and farming techniques. It becomes challenging for the farmers to satisfy our planet's evolving requirements and the expectations of merchants, customers, etc. Some of the challenges the farmers face are- (i) Dealing with climatic changes because of soil erosion and industry emissions (ii) Nutrient deficiency in the soil, caused by a shortage of crucial minerals such as potassium, nitrogen, and phosphorus can result in reduced crop growth. (iii) Farmers make a mistake by cultivating the same crops year after year without experimenting with different varieties. They add fertilizers randomly without understanding the inferior quality or quantity. The paper aims to discover the best model for crop prediction, which can help farmers decide the type of crop to grow based on the climatic conditions and nutrients present in the soil. This paper compares popular algorithms such as K-Nearest Neighbor (KNN), Decision Tree, and Random Forest Classifier using two different criterions Gini and Entropy. Results reveal that Random Forest gives the highest accuracy among the three.

1. Introduction

Machine learning is a valuable decision-making tool for predicting agricultural yields and deciding the type of crops to sow and things to do during the crop growing season. In order to aid crop prediction studies, several machine learning methods have been used.

Machine learning techniques are utilized in various sectors, from evaluating customer behavior in supermarkets to predicting customer phone usage. For some years, agriculture has been using machine learning techniques. Crop prediction is one of agriculture's complex challenges, and several models have been developed and proven so far. Because crop production is affected by many factors such as atmospheric conditions, type of fertilizer, soil, and seed, this challenge necessitates using several datasets. This implies that predicting agricultural productivity is not a simple process; rather, it entails a series of complicated procedures. Crop yield prediction methods can now reasonably approximate the actual yield, although more excellent yield prediction performance is still desired.

The project aims to compare various supervised learning algorithms like KNN, Decision Tree, and Random Forest on the dataset containing 22 varieties of crops. For the Decision Tree and Random Forest Classifier, the model's performance is calculated under two criterions- Entropy and Gini Index. The results reveal that the suggested machine learning technique's effectiveness is compared to the best accuracy with precision, recall, and F1 Score.

2. Literature Survey

Given the significance of crop prediction, numerous suggestions have been proposed in the past with the goal of improving crop prediction accuracy. In this paper feed-forward back propagation Artificial Neural Network methodology has been approached to model and forecast various crop yields at rural areas based on parameters of soil(PH, nitrogen, potassium, etc.) and parameters related to the atmosphere (rainfall, humidity, etc.) [1].

This paper looks at five of Tamil Nadu's most important crops- rice, maize, ragi, sugarcane, and tapioca during a five-year period beginning in 2005. [2]. In order to get the maximum crop productivity, various factors such as rainfall, groundwater, and cultivation area, and soil type were used in the analysis. K-Means technique was used for the clustering, and for the classification, the study looked at three different types of algorithms: fuzzy, KNN, and Modified KNN. After the analysis, MKNN gave the best prediction result of the three algorithms.

An application for farmers can be created that will aid in the reduction of many problems in the agriculture sector [3]. In this application, farmers perform single/multiple testing by providing input such as crop name, season, and location. As soon as one provides the input, the user can choose a method and mine the outputs. The outputs will show you the crop's yield rate. The findings of the previous year's data are included in the datasets and transformed into a supported format. The machine learning models used are Naïve Bayes and KNN.

To create the dataset, information about crops over the previous ten years was gathered from a variety of sources, such as government websites. An IoT device was setup to collect the atmospheric data using the components like Soil sensors, Dht11 sensor for humidity and temperature, and Arduino Uno with Atmega as a processor. Naive Bayes, a supervised learning algorithm obtaining an accuracy of 97% was further improved by using boosting algorithm, which makes use of weak rule by an iterative process to bring higher accuracy [5]. To anticipate the yield, the study employs advanced regression techniques such as ENet, Kernel Ridge, and Lasso algorithms [4]. The three regression techniques are improved by using Stacking Regression for better prediction.

However, when a comparison study is conducted between the existing system and the proposed system employing Naive Bayes and Random Forest, respectively. The proposed system comes out on top. Because it is a bagging method, the random forest algorithm has a high accuracy level, but the Naïve Bayes classifier's accuracy level is lower as the algorithm is probability based. [6].

This paper contributes to the following aspects- (a) Crop production prediction utilizing a range of Machine Learning approaches and a comparison of error rate and accuracy for certain regions. (b) An easy-to-use mobile app that recommends the most gainful crop. (c) A GPS-based location identifier that can be used to obtain rainfall estimates for a specific location. (d) A system that recommends the prime time to apply fertilizers [7]. On the given datasets from Karnataka and Maharashtra, different machine learning algorithms such as KNN, SVM, MLR, Random Forest, and ANN were deployed and assessed for yield to accuracy [9]. The accuracy of the above algorithms is compared. The results show that Decision Tree is the most accurate of the standard algorithms used on the given datasets, with a 99.87% accuracy rate.

Regression Analysis is applied to determine the relationship between the three factors: Area Under Cultivation, Food Price Index, and Annual Rainfall and their impact on crop yield. The above three factors are taken as independent variables, and for the dependent variable, crop yield is taken into consideration. The R^2 obtained after the implementation of RA shows these three factors showed slight differences indicating their impact on the crop yield [8].

In the proposed paper, the dataset is collected from the government websites such as APMC website, VC Farm Mandya, which contains data related to climatic conditions and soil nutrients [10]. Two machine learning models were used; the model was trained using the Support Vector Machine model with Radial Basis Function kernel for rainfall prediction and Decision Tree for the crop prediction.

A comparative study of various machine learning can be applied on a dataset with a view to determine the best performing methodology. The prediction is found by applying the Regression Based Techniques such as Linear, Random Forest, Decision Tree, Gradient Boosting, Polynomial and

Ridge on the dataset containing details about the types of crops, different states, and climatic conditions under different seasons [11]. The parameters used to estimate the efficiency of these techniques were mean absolute error, root mean square error, mean squared error, R-square, and cross validation. Gradient Boosting gave the best accuracy- 87.9% for the target variable 'Yield' and Random Forest- 98.9% gave the best accuracy for the target value 'Production'.

The DHT22 sensor is recommended for monitoring live temperature and humidity [12]. The surrounding air is measured with a thermistor and a capacitive humidity sensor and outputs a digital signal on the data pin to the Arduino Uno port pin. The humidity value ranges from 0-100% RH and -40 to 80 degrees Celsius to read the temperature. The above two parameters and soil characteristics are considered as input to three different machine learning models: Support Vector Machine, Decision Tree, and KNN. The Decision Tree gave better accuracy results.

Table 1. Approach to Crop Prediction

Author	Proposed Model	Accuracy
M.Kalimuthu et.al(2020)[5]	Naïve Bayes	97%
V. Geetha et.al(2020)[6]	Naïve Bayes and Random Forest Classifier	95% Random Forest Classifier
Shilpa Mangesh P et.al(2021)[7]	Support Vector Machine, K-Nearest Neighbor, Multivariate Linear Regression, Artificial Neural Network, Random Forest,	95% Random Forest Classifier
S Bharath et.al(2020)[9]	Support Vector Machine, Decision Tree, K-Nearest Neighbor, Random Forest	99.87% Decision Tree Classifier
Payal Gulati and Suman Kumar(2020)[11]	Linear Regression, Random Forest, Decision Tree, Gradient Boosting Regression, Ridge Regression, Polynomial Regression	98.9% Random Forest Classifier
Archana Gupta et.al(2020)[12]	K-Nearest Neighbor, Support Vector Machine, Decision Tree Classifier	91.03% Decision Tree Classifier

3. Proposed Work

3.1. K-Nearest Neighbor Classifier

The K-Nearest Neighbor algorithm is based on the supervised learning technique and is a simple machine learning algorithm. The K-NN technique saves all possible and classifies the incoming data depending on how similar they are to the actual data. This means that the K-NN technique can swiftly classify new instances into a precisely defined category. The KNN technique can be used in both regression and classification problems but, it is most likely to be used in classification.

KNN technique has two properties. First, the model is based on the dataset or, it is not required to identify parameters for the distribution. Hence it is referred to as non-parametric. Second, there is no learning taking place; instead, it just stores the training data. The classification of the dataset happens during the testing phase, due to which the testing phase becomes time-consuming and takes a lot of memory. This property is known as lazy-learner.

3.1.1. KNN Algorithm. Step 1: K, i.e., the number of neighbors is selected. The primary deciding factor is the number of neighbors.

Step 2: Using distance measures, determine the distance between two points like Euclidean distance

$$Euclidean\ distance = d(b, a) = \sqrt{\sum_{i=1}^{n}(b_i - a_i)^2}$$

Step 3: K nearest neighbors are taken into account according to the calculated Euclidean distance.
Step 4: Figure the number of data points in each class surrounded by these K neighbors.
Step 5: The class with the highest number of neighbors is assigned to the new data points.
Step 6: The label is voted on, and the model is ready.

3.2. Decision Tree Classifier

Decision Tree is a supervised learning technique used for both classification and regression problems where each path is a set of decisions leading to a class. A sequence of questions is asked by taking an instance from the training set. The non-terminal node such as root and internal nodes has decision attributes. The decision is made by comparing the instance with the decision attribute, resulting in the split and jumping to the next node. This splitting continues, generating sub-trees until it reaches a leaf node which determines class labels for that instance. It divides the tree recursively, which is known as recursive partitioning. With high accuracy, decision trees are capable of handling high-dimensional data. It's a flowchart diagram-style representation that closely parallels human-level thinking. As a result, decision trees are simple to explain and apprehend.

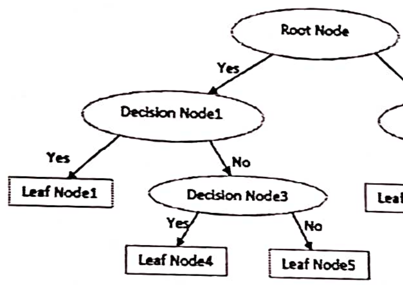

Figure 1. Decision Tree Classifier

3.2.1. Decision Tree Algorithm. Step 1: Starting with the root node of the tree, which consists of the entire dataset, says S.
Step 2: The most appropriate attribute is obtained from the dataset by applying the Attribute Selection Measure (ASM).
Step 3: The S is divided into subdivisions that enclose feasible values for the most appropriate attributes.
Step 4: The node is formed in the decision tree with the most appropriate attribute.
Step 5: The tree formation is setup by iteratively repeating this method for each child until one of the following requirements is met:

- The tuples are entirely correlated with the same attribute value.
- There are no further attributes accessible.
- There aren't any more instances.

3.2.2. Steps to Split. The dataset used in the project has numerical values. The decision tree works with the numerical values in the following ways:
Step 1: Sorting all the values.
Step 2: It will consider a threshold value from the feature values.
Step 3: Feature value will split into two parts such that the left node contains feature values less than a threshold value, and the right node contains feature values greater than a threshold value.
Step 4: The next feature value will consider as a threshold value and again create the same split as Step 3.

Step 5: Entropy/Gini and Information Gain are calculated of each split, and from the two splits, the split with better information gain is considered.

$$I(Attribute) = \frac{\sum p_i + n_i}{p + n}$$

Where p_i denotes the number of yes values; n_i denotes the number of no values for that particular attribute; p and n are the numbers of yeses and noes of the entire sample, respectively.

$$Information\ Gain = Entropy(S) - I(Attribute)$$

Where Entropy(S) denotes entropy of sample S; I(Attribute) denotes Average Information of the particular attribute.

Step 6: Repeat from Step 2 to Step 5. In this way, it will get branches for the decision tree.

3.2.3. Entropy and Gini Index. The criteria for measuring Information Gain are the Gini index and Entropy. Information gain is a measurement of how much information is gained about an attribute and the reduction in entropy. Entropy and Gini Index are the metrics that measure the impurity of the nodes. A node is considered as impure if it has multiple classes else, it is considered as pure.

Entropy is a metric that gives the degree of impurity in a specified attribute. The following formula can be used to compute entropy:

$$Entropy(S) = -P(yes)\ log_2\ P(yes) - P(no)\ log2\ P(no)$$

$$Entropy(S) = \sum_{i=1}^{n} -p_i log_2 p_i$$

Where S denotes complete sample, P(yes) denotes the probability of yes; P(no) denotes the probability of no.

Gini Index: Gini is estimated by deducting the sum of squared probabilities of each class from one. The lower Gini Index value is preferred rather than a higher value. Scikit-learn takes "Gini" as the default value and supports "Gini" criteria for Gini Index.

$$Gini\ Index = 1 - \sum_{i=1}^{c} (P_i^2)$$

Where P_i denotes the probability of a tuple, say R belonging to class C_i

3.3. Random Forest Classifier

The Random Forest method consists of multiple decision tree classifiers to enhance the model's performance. Here ensemble learning is the principle used to resolve complicated problems. It is a supervised learning algorithm. Decision trees are created at random using the instances from the training set. Each of the decision trees gives out predictions as their outcome. The final prediction for the model is decided by majority voting. One of the reasons for its popularity as a machine learning approach is that it can handle the issue of overfitting, and accuracy can be increased by using more trees.

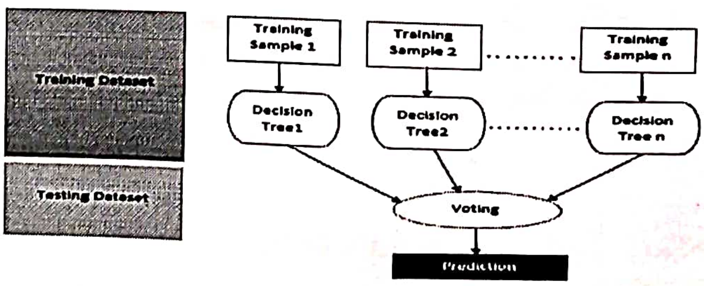

Figure 2. Random Forest Classifier

3.3.1. Random Forest Algorithm. Step 1: K instances are chosen at random from the given training dataset.
Step 2: Decision trees are created for the chosen instances.
Step 3: The N is selected for the number of estimators to be created.
Step 4: Step 1 & Step 2 is repeated.
Step 5: For the new instance, the predictions of each estimator is determined, and the category with the highest vote is assigned.

4. Methodology

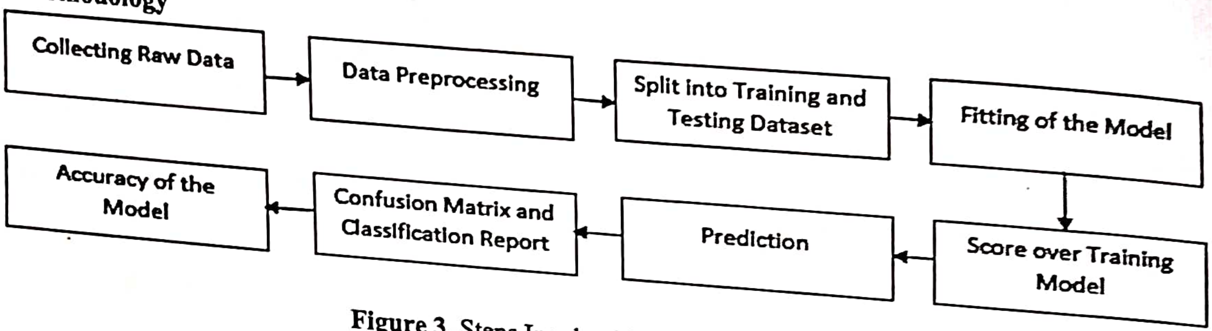

Figure 3. Steps Involved in the Methodology

4.1. Collecting the Raw Data

The practice of cumulating and scrutinizing data from different sources is known as data collection. Data collection is a way to keep track of past occurrences so that one can utilize data analysis to detect repetitive patterns. The 'Crop Recommendation' dataset is collected from the Kaggle website. The dataset takes into account 22 different crops as class labels and 7 features- (i) Nitrogen content ratio (N) (ii) Phosphorus content ratio (P) (iii) Potassium content ratio (K) in the soil, (iv) Temperature expressed in degree Celsius (v) Percentage of Relative Humidity (vi) ph value and (vii) Rainfall measured in millimeters.

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice
...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee

Figure 4. Dataset Sample

4.2. Data Preprocessing

The process of modifying raw data into a form that analysts and data scientists can use in machine learning algorithms to find insights or forecast outcomes is called Data preprocessing. In this project, the data processing method is to find missing values. Getting every data point for every record in a dataset is tough. Empty cells, values like null or a specific character, such as a question mark, might all indicate that data is missing. The dataset used in the project didn't have any missing values.

4.3. Train and Test Split
It is a process of splitting the dataset into a training dataset and testing dataset using train_test_split() method of scikit learn module. 2200 data in the dataset has been divided as 80% of a dataset into training dataset-1760 and 20% of a dataset into testing dataset-440 data.

4.4. Fitting the model
Modifying the model's parameters to increase accuracy is referred to as fitting. To construct a machine learning model, an algorithm is performed on data for which the target variable is known. The model's accuracy is determined by comparing the model's outputs to the target variable's actual, observed values. Model fitting is the ability of a machine learning model to generalize data comparable to that with which it was trained. When given unknown inputs, a good model fit refers to a model that properly approximates the output.

4.5. Checking the score over a training dataset
Scoring, often known as prediction, is the act of creating values from new input data using a trained machine learning model. Using model.score() method calculating the score of each model over a training dataset shows how well the model has learned.

4.6. Predicting the model
When forecasting the likelihood of a specific result, "prediction" refers to the outcome of an algorithm after it has been trained on a previous dataset and applied to new data. Predicting the model using predict() method using test feature dataset. It has given the output as an array of predicted values.

4.7. Confusion Matrix and Classification Report
Confusion Matrix and Classification Report are the methods imported from the metrics module in the scikit learn library are calculated using the actual labels of test datasets and predicted values.

Confusion Matrix gives the matrix of frequency of true negatives, false negatives, true positives and false positives.

Classification Report is a metric used for evaluating the performance of a classification algorithm's predictions. It gives three things: Precision, Recall and f1-score of the model.

Precision refers to a classifier's ability to identify the number of positive predictions which are relatively correct. It is calculated as the ratio of true positives to the sum of true and false positives for each class.

$$Precision = \frac{TP}{TP + FP}$$

Where Precision-Positive Prediction Accuracy; TP-True Positive; FP-False Positive

Recall is the capability of a classifier to discover all positive cases from the confusion matrix. It is calculated as the ratio of true positives to the sum of true positives and false negatives for each class.

$$Recall = \frac{TP}{TP + FN}$$

Where Recall- The percentage of positives that were correctly identified; FN-False Negative

F1 score is a weighted harmonic mean of precision and recall, with 0.0 being the worst and 1.0 being the best. Since precision and recall are used in the computation, F1 scores are often lower than accuracy measurements.

$$F1\,score = \frac{2 * PR}{(P + R)}$$

Where P-Precision; R-Recall

4.8. Accuracy

The number of correct predictions divided by the total number of predictions is known as model accuracy. The accuracy of the model is calculated using accuracy_score() method of scikit learn metrics module.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP-True Positive; FP-False Positive; TN-True Negative; FN-False Negative.

5. Results

5.1. *KNN Classifier Predictions*

In the model, we have chosen neighbors K=5. Increasing or decreasing the value of K will give less accuracy to the model.

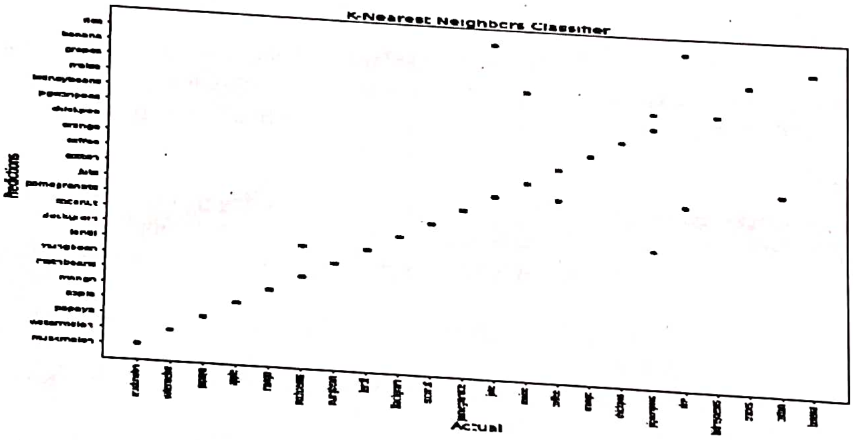

Figure 5. KNN Prediction-comparison of predicted values and actual values

As observed from figure 5, the points along the straight line depict the correct predictions, and the points outside the straight line are wrong predictions. For example, the actual value should be pigeon peas, but it has been predicted as blackgram, which is a false prediction.

5.2. *Decision Tree Classifier Prediction of Entropy and Gini Criterion*

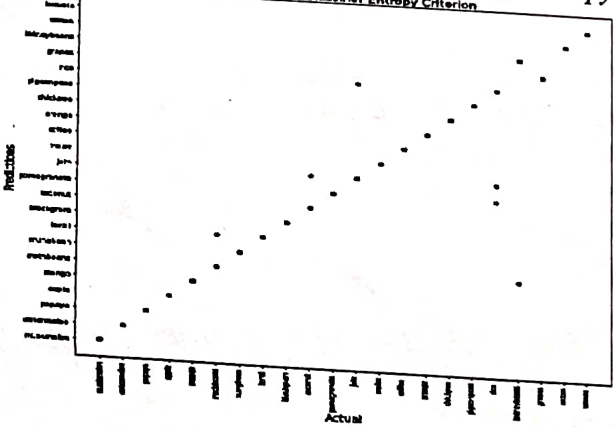

Figure 6. Decision Tree Entropy Criterion Prediction-comparison of predicted values and actual values.

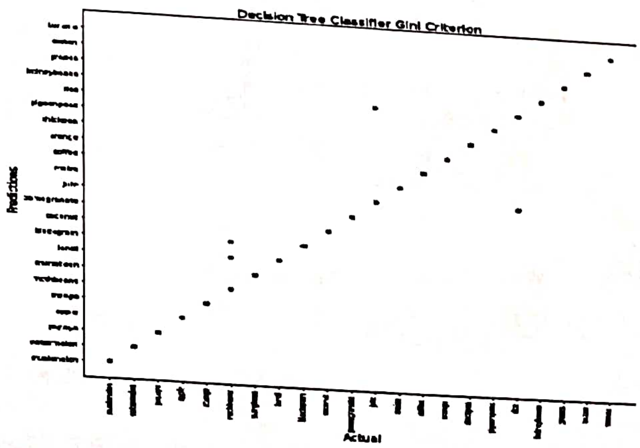

Figure 7. Decision Tree Gini Criterion Prediction-comparison of predicted values and actual values.

As observed from the above figure 6 & figure 7, Decision Tree Gini Criterion has less number of wrong predictions compared to Decision Tree Entropy Criterion. Thus Gini showing better accuracy than Entropy.

5.3. Random Forest Classifier Prediction of Entropy and Gini Criterion

While designing the model, the number of estimators chosen is 100. It means our model is designed with 100 decision trees. The increase in the number of estimators will not affect the accuracy of the model as it gives the best result with 100 estimators.

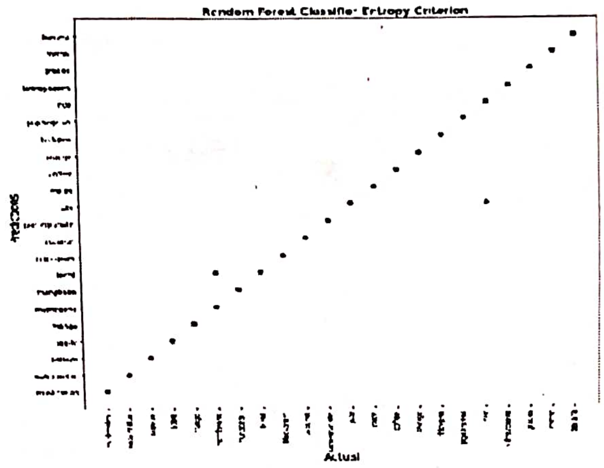

Figure 8. Random Forest Entropy Criterion Prediction-comparison of predicted values and actual values

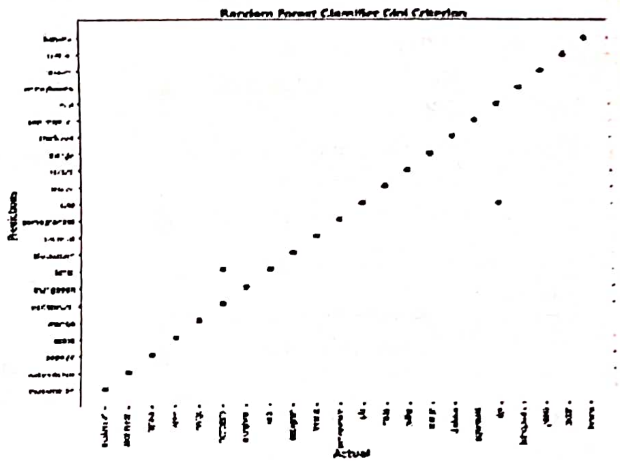

Figure 9. Random Forest Gini Criterion Prediction-comparison of predicted values and actual values

As observed from figure 8 & figure 9, the Random Forest shows the same predictions for both Entropy and Gini Index Criterion.

5.4. Accuracy Comparison of all models

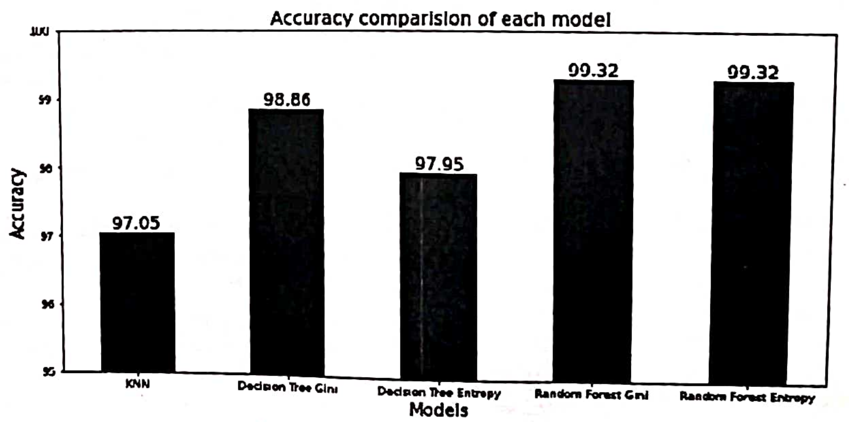

Figure 10. Accuracy of all models

Table 2. Comparison of Training Score and Model Accuracy

Models	Training Score	Model Accuracy
K-Nearest Neighbor Classifier	98.97	97.04
Decision Tree Classifier Entropy Criterion	100.0	97.95
Decision Tree Classifier Gini Criterion	100.0	98.86
Random Forest Classifier Entropy Criterion	100.0	99.32
Random Forest Classifier Gini Criterion	100.0	99.32

6. Conclusion and Future Work

The comparative study of three different supervised machine learning models (KNN, Decision Tree, and Random Forest) is done to predict the best-suited crop for the particular land that can help farmers to grow crops more efficiently. In completion, we concluded that the crop prediction dataset showed the best accuracy with Random Forest Classifier both in Entropy and Gini Criterion with 99.32%. In contrast, K-Nearest Neighbor has the lowest accuracy among the three with 97.04%, and the accuracy of Decision Tree Classifier is in between KNN and Random Forest Classifier. When comparing the accuracy value, Decision Tree Gini criterion gave a better accuracy of 98.86% compared to Decision Tree Entropy Criterion. In the future, new data from the fields can be collected to get a clear image of the soil and incorporate other machine learning algorithms and deep learning algorithms such as ANN or CNN to classify more varieties of crops.

References

[1] Dahikar S and Rode S V 2014 Agricultural crop yield prediction using artificial neural network approach *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* **vol 2** Issue 1 pp 683-6.

[2] Suresh A, Ganesh P and Ramalatha M 2018 Prediction of major crop yields of Tamilnadu using K-means and Modified KNN *2018 3rd International Conference on Communication and Electronics Systems (ICCES)* pp 88-93 doi: 10.1109/CESYS.2018.8723956.

[3] Medar R, Rajpurohit V S and Shweta S 2019 Crop yield prediction using machine learning techniques *IEEE 5th International Conference for Convergence in Technology* (I2CT) pp 1-5 doi: 10.1109/I2CT45611.2019.9033611.

[4] Nishant P S, Venkat P S, Avinash B L and Jabber B 2020 Crop yield prediction based on Indian agriculture using machine learning *2020 International Conference for Emerging Technology* (INCET) pp 1-4 doi: 10.1109/INCET49848.2020.9154036.

[5] 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.

[6] Geetha V, Punitha A, Abarna M, Akshaya M, Illakiya S and Janani A P 2020 An effective crop prediction using random forest algorithm 2020 *International Conference on System, Computation, Automation and Networking* (ICSCAN) pp 1-5 doi: 10.1109/ICSCAN49426.2020.9262311.

[7] Pande S M, Ramesh P K, Anmol A, Aishwaraya B R, Rohilla K and Shaurya K 2021 Crop recommender system using machine learning approach 2021 *5th International Conference on Computing Methodologies and Communication* (ICCMC) pp 1066-71 doi: 10.1109/ICCMC51019.2021.9418351.

[8] Sellam V, and Poovammal E 2016 Prediction of crop yield using regression analysis *Indian*