Journal of Basics and Applied Sciences Research (JOBASR)
ISSN print: 3026-9091
Volume 2(1) March 2024
DOI: https://doi.org/10.33003/jobasr-2024-v2i1-43

Crop Recommendation Predictive Analysis using Ensembling Techniques

Muhammad Umar Abdullahi[1]*, Morufu Olalere[2], Gilbert I. O. Aimufua[2], Kene Tochukwu Anyachebelu[2] and Bako Halilu Egga[2]
[1]Department of Computer Science, Federal University of Technology, Owerri, Nigeria
[2]Department of Computer Science, Nasarawa State University, Keffi, Nigeria
*Corresponding Author Email: umarfaruq54@gmail.com

ABSTRACT
Crop recommendation systems play a crucial role in modern agriculture by aiding farmers in making well-informed choices to optimize crop yield and resource utilization. Ensemble learning approaches can significantly improve the effectiveness of crop recommendation systems. To achieve this, multiple forecasts are combined from various models. In this paper, a complete Machine Learning Pipeline is used to evaluate the performance of ensemble learning models in crop recommendation tasks. A diverse dataset is used to select and train four ensemble learning methods, Bagging, Voting, Stacking, and One-Vs-Rest (OVR), as separate classifiers. The dataset includes various agricultural factors such as soil characteristics, meteorological conditions, and past crop productivity. Various metrics, including accuracy, precision, recall, F1-score, and support, are utilized for each model. Bagging is considered the most effective ensemble learning technique, demonstrating excellent levels of accuracy and overall performance. The bagging algorithm achieves a high level of accuracy, reaching 99.32%. It also achieves perfect precision, recall, and F1-score metrics, with values of 0.99, 1.00, and 1.00 respectively. The support value, which represents the number of instances used for evaluation, is 141. This study provides valuable perspectives on the choice of appropriate ensemble learning models for crop recommendation tasks. Consequently, it enables farmers and other individuals involved in agriculture to make well-informed choices using data, resulting in enhanced agricultural output and sustainability.

Keywords:
Comparative Analysis,
Ensembling Learning Models,
Crop recommendation,
Evaluation Metrics,
Model Assessment.

INTRODUCTION

Agriculture serves as the fundamental support for numerous economies, supplying sustenance, livelihoods, and basic resources for diverse sectors. By 2050, the world population is estimated to reach around 10 billion, leading to a substantial rise in the need for food (Tripathi, Mishra, Maurya, Singh & Wilson, 2019).. This would need advancements in agricultural production and efficiency (Serraj, Krishnan & Pingali, 2019; Falcon, Naylor & Shankar, 2022). Technology plays a significant role in modern agriculture, particularly with the progress in data science and machine learning. These innovations provide new opportunities for improving agricultural methods.

Historically, farmers have traditionally depended on practical knowledge and regional experience to make informed choices on crop selection, timing of planting, and allocation of resources (Chakauya, Materechera, Jiri, Chakauya & Machete, 2023). Nevertheless, these approaches are frequently constrained by subjectivity, fluctuations in environmental circumstances, and the incapacity to efficiently handle and interpret substantial amounts of data (Limpo et al., 2022; Kom, Nethengwe, Mpandeli & Chikoore, 2022). Consequently, there has been an increasing interest in utilizing technology to provide data-driven methods for making decisions in agriculture.

Machine learning, also known as artificial intelligence, has become a powerful tool for analysing agricultural data and generating valuable insights (Redhu, Thakur, Yashveer & Mor, 2022). Machine learning models may utilize algorithms to analyse various datasets, such as soil composition, weather patterns, satellite images, and historical agricultural production data. Through this analysis, these models can find intricate patterns and linkages that may not be easily observable by humans (Shaikh, Rasool & Lone, 2022). This feature allows for the creation of crop recommendation systems that can forecast the optimal crops for a certain region by considering many parameters.

Ensemble learning is the process of amalgamating predictions from numerous base models to provide a solitary and more precise forecast (Mohammed & Kora, 2023). Ensembling approaches can enhance overall performance by harnessing the variety of various models, therefore mitigating the limitations of individual classifiers (Ganaie, Hu, Malik, Tanveer & Suganthan, 2022). Stacking, Bagging, Voting, and One-Vs-Rest (OVR) are often employed ensemble learning techniques, each employing a distinct strategy for aggregating predictions (Zounemat-Kermani, Batelaan, Fadaee & Hinkelmann, 2021).

While there is a growing interest in employing ensemble learning for agricultural purposes, there is a lack of comprehensive comparative assessments to determine the most effective approach for crop recommendation tasks (Agrawal, Govil, & Kumar, 2024). This paper aims to address this gap by conducting a thorough analysis of the efficacy of different ensembling procedures using real agricultural data. This research tends to assess the precision, resilience, and computational efficiency of ensemble learning techniques, including Stacking, Bagging, Voting, and OVR. The aim is to get valuable information regarding the efficacy of these methodologies for crop recommendation (Sajitha, Andrushia, Anand & Naser, 2024).

Furthermore, the paper will examine the impact of various dataset factors, such as size, dimensionality, and class distribution, on the efficacy of ensembling algorithms. A thorough understanding of how these factors influence the effectiveness of ensemble learning might offer useful insights for the development and implementation of more robust crop recommendation systems tailored to different agricultural contexts (Shams, Gamel & Talaat, 2024).

The paper enhances agricultural technology by offering empirical data on the effectiveness of ensemble learning approaches for crop recommendation. The results of this study can provide valuable insights for improving the accuracy and dependability of crop recommendation systems. This, in turn, can enable farmers to make informed decisions based on data, leading to increased productivity, sustainability, and resilience in agriculture.

MATERIALS AND METHODS

The paper utilizes the Machine Learning Pipeline or Data Science Workflow methodology (Biswas, Wardat & Rajan, 2022). The methodology has eight sequential steps. The procedure illustrated in Figure 1 comprises many stages: data collection, data exploration, pre-processing, dataset partitioning, model training, model testing, performance evaluation, performance comparison, and conclusion (Biswas, Wardat & Rajan, 2022). This figure was modified from the original work of Rana et al. (2015) to align with the research approach utilized in this paper. The attainment of this target was achieved by the systematic implementation of a well-structured series of procedures.

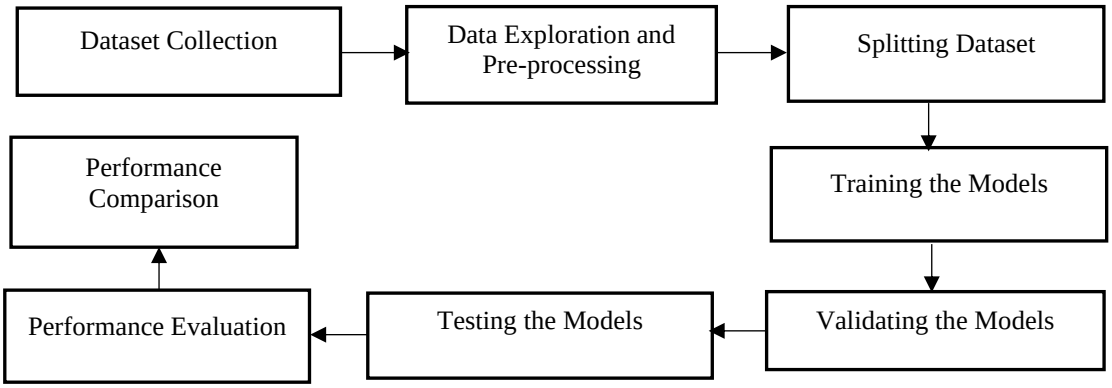

Figure 1: Data Science Workflow
Source: Author Based on (Rana *et al*., 2015)

Dataset Collection

The initial step involved the collection of the dataset in CSV format, followed by its importation into Jupyter Notebook for further analysis.

Data Exploration and Pre-processing

This phase encompassed four key stages. First, a thorough exploration of the data was conducted to understand its characteristics. Subsequently, features with negative correlations to the target variable were identified and removed to enhance model performance. Finally, feature scaling was applied to ensure that all features were on a consistent scale.

Splitting the Dataset

Following the importation of the dataset, it was divided into two distinct subsets. The first subset was utilized for training the machine learning models, while the second

subset was employed to evaluate the performance of these models. This step was undertaken to assess the performance of the models on an independent dataset, therefore mitigating the risk of overfitting.

Training the Models

During the training phase, seven separate ensembling learning models were utilized, specifically Stacking, Bagging, Voting, and One-Vs-Rest (OVR). The aforementioned models were employed to construct prediction algorithms.

Validation of Models

To improve the prediction performance of the machine learning models, a validation set was utilized. This particular collection played a crucial role in the process of adjusting model parameters to maximize their performance.

Testing the Models

The models, having undergone training and validation, were rigorously tested using the testing dataset to assess their accuracy and predictive power.

Performance Evaluation

To gauge the effectiveness of the models, a thorough performance evaluation was conducted. This evaluation employed metrics such as the scikit-learn accuracy score and confusion matrix.

Performance Comparison

The next phase was a thorough comparison of the seven machine learning models. The goal of this comparison research was to determine which model outperformed the others in the essential duty of predicting crops, therefore adding vital insights to the area of medical diagnostics.

Source of Dataset

The dataset was obtained from https://www.kaggle.com/datasets/aksahaha/crop-recommendation and includes data on the amounts of nitrogen, phosphorus, and potassium in the soil as well as measurements of temperature, humidity, pH, and rainfall and how they affect the growth of crops (Nti *et al.*, 2023). This dataset can be utilized to create data-based suggestions for achieving the best possible nutrient and environmental conditions to enhance crop yield (Gosai *et al*., 2021). The data size was 2200 records and seven predictors (Muhammed, Ahvar, Ahvar & Trocan, 2023). The target variable consists of twenty-two classes representing different crops (i.e., 'mungbean', 'apple', 'kidney-beans', 'banana' 'maize', 'blackgram', 'chickpea', 'mothbeans', 'coconut', 'coffee', 'cotton', 'grapes', 'jute', 'pigeonpeas', 'papaya', 'mango', 'lentil', 'muskmelon', 'orange', 'watermelon', 'pomegranate' and 'rice') each with one hundred (100) samples.

Ensemble Learning Techniques

Ensemble methods encompass various techniques, with common approaches including Bagging (Bootstrap Aggregating), Boosting, Stacking, and OVR (Jia, Liang & Liang, 2023).

Bagging (Bootstrap Aggregating)

Bagging is a technique where several models are trained separately on various subsets of the dataset (Breiman, 1996; González, García, Del-Ser, Rokach & Herrera, 2020; Malek et al.,2023). The Bagging method involves the creation of subsets via bootstrapping, which is a resampling strategy where instances are randomly picked with replacements from the original dataset (Gul, Mashwani, Aamir, Aldahmani & Khan, 2023). As a consequence, this leads to a variety of subsets that are used to train separate models inside the ensemble. Bagging, a technique introduced by Nosrati and Rahmani in 2023, aims to reduce the influence of noise or outliers in the data by training models independently on different subsets, thereby introducing variety. An exemplary instance of Bagging is the Random Forest method. In a Random Forest, several decision trees are trained using distinct bootstrap samples of the dataset (Sun et al.,2023). During the prediction process, each tree in the forest provides its own prognosis individually. The final prediction is then derived by combining these individual predictions (Becker, Geubbelmans, Rousseau, Valkenborg & Burzykowski, 2024).

The utilization of an ensemble technique frequently results in a model that is more robust and precise, with the ability to effectively apply its knowledge to unfamiliar data (Jitpakdeebodin & Sinapiromsaran, 2023). Bagging improves the predictive performance of an ensemble by decreasing overfitting and variance. It is an important approach in machine learning, especially when dealing with complicated and noisy data that may affect the performance of individual models.

Boosting

Boosting is a technique that involves training models in a sequential manner to correct the errors made by previous models (Emami & Martínez-Muñoz, 2023; Kunapuli, 2023). The core concept of boosting is to progressively enhance the predicted performance of the model by assigning greater importance to difficult data points (Attri, Awasthi, Sharma & Rathee, 2023). AdaBoost and Gradient Boosting are prominent algorithms in the boosting family. AdaBoost stands for Adaptive Boosting, while Gradient Boosting was developed by Thotad, Bharamagoudar, and Kallur in 2023.

During the boosting process, weak learners, which are often basic decision trees known as "stumps," are taught one after another in a sequential manner (Avramopoulos & Vasiloglou, 2023).

Following each iteration, greater emphasis is placed on misclassified instances, encouraging the succeeding learner to concentrate on the previously difficult aspects of the material (Miller et al. 2023). This iterative process of adjusting weights allows the model to continuously improve its capacity to identify intricate patterns and connections within the dataset.

In boosting, the ultimate forecast is usually obtained by combining the weak learners' predictions using weights that reflect their accuracy during the training phase (Demir & Sahin, 2023). Boosting is renowned for its ability to enhance model performance, particularly in scenarios where individual models may encounter difficulties.

Stacking

Stacking, which is also known as Stacked Generalization, is a technique that combines predictions from many models using a meta-model (Seireg, Omar & Elmahalawy, 2023). During the stacking process, a set of base models is trained individually. Each of these models captures certain elements or makes unique mistakes on the training data (de-Zarzà, de-Curtò, Hernández-Orallo & Calafate, 2023). The forecasts produced by these foundational models are subsequently combined, and a meta-model, often a less complex algorithm such as linear regression, is trained using these forecasts (Chen, Zeb, Nanehkaran, & Zhang, 2023). The meta-model is designed to optimize the combination of outputs from the basis models, resulting in a more precise and accurate prediction during the testing or validation phase (Liang & Liu, 2023).

The fundamental principle of stacking is its capacity to use the synergistic advantages of different models, therefore exceeding the predictive capabilities of any one model (Yang et al., 2023). Although the concept of stacking shows potential for enhancing performance, it is important to carefully evaluate the variety of models used and select the most suitable meta-model in order to achieve the best possible outcomes in practical applications. Stacking is a powerful technique in ensemble learning that excels in complicated tasks, providing better abilities for subtle pattern detection and improved overall accuracy.

One-Vs-Rest (OVR)

The one-versus-rest (OVR) approach, also known as one-versus-all, is a popular technique used in multi-class classification tasks. It involves training a binary classifier for each class, where the positive class is the one being trained, and all other classes are the negative class (Hsu & Lin, 2002; Rifkin & Klautau, 2004).

The OVR approach involves training a binary classifier for each class in the dataset distinctly. During the training process, examples that belong to the positive class are labeled as such, while instances from all other classes are gathered together and labeled as the negative class. Each binary classifier is trained to differentiate its assigned class from all other classes, resulting in a collection of classifiers specifically designed to identify each individual class (Crammer & Singer, 2002).

During the prediction process, each binary classifier is provided with the instance, and the predicted class for that instance is determined by assigning it the class with the highest confidence score or probability. The final prediction is determined by the classifier that has the highest confidence for its allocated class (Frenay & Verleysen, 2014).

An inherent benefit of the OVR method is its straightforwardness and flexibility to be easily expanded. OVR, or one-vs-rest, is a technique that breaks down the multi-class classification issue into several binary classification subproblems. This approach enables the use of common binary classifiers like Support Vector Machines (SVMs) or Logistic Regression without the need for any changes to the underlying methods (Hsu & Lin, 2002).

Furthermore, OVR can be highly efficient in addressing unbalanced class distributions by prioritizing the differentiation of each class separately rather than taking into account the overall class distribution as a whole (Rifkin & Klautau, 2004). In addition, OVR (One-vs-Rest) classification tends to get good results in situations when classes are not mutually exclusive or when there is a substantial overlap across class boundaries (Hsu & Lin, 2002).

Performance Metrics for Classification

The evaluation criteria utilized for gauging the effectiveness of this analysis are as follows:

Accuracy

The effectiveness of a model is evaluated based on the ratio of correct predictions generated for various types of forecasts (Ricciardi, Ramankutty, Mehrabi, Jarvis & Chookolingo, 2018). The assessment technique entails evaluating the precision of categorization by comparing the number of accurately classified cases to the total number of occurrences (Petropoulos & Siemsen, 2023). The assessment of accuracy is especially useful in situations when the distribution of classes in the target variable is evenly distributed across the dataset. Equation 1 represents this concept.

$$ACCURACY = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

Sensitivity or Recall

Sensitivity is a quantitative measure used to determine the percentage of positive scenarios that were accurately recognized by the model, but mistakenly classified as negative. Occasionally referred to as recall or true positive rate, as stated by Hutter (2012). Mathematically, it is defined as the quotient obtained by dividing the number of true positive (TP) occurrences by the total of true positive and false negative (FN) cases. The mathematical expression is as follows:

$$SENSITIVITY = \frac{TP}{TP+FN} \quad (2)$$

Specificity

Specificity, sometimes referred to as the true negative rate, is an important concept in the field of software defect analysis (Everitt, Goertzel & Potapov, 2017). Equation 3 quantifies the proportion of defect-free occurrences in the software system that are accurately identified as such by the model.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \quad (3)$$

Detection Rate

The detection rate is the ratio of successfully detected events to the total sample size (Flasiński, 2016). This statistic measures the efficiency of accurately identifying instances within the dataset.

F1 score rate

The F1 score is a metric that calculates the weighted average of accuracy and recall (Bach, 2020). Consequently, this score considers the equilibrium between incorrect positive results and incorrect negative results.

$$\text{F1 Score} = 2 * \frac{precision*recall}{precision+recall} \quad (6)$$

Precision

Precision is a quantitative measure that evaluates the correctness of positive predictions generated by a model (Davis, 2015). Precision is the quotient obtained by dividing the number of accurately anticipated positive samples by the total number of samples predicted as positive.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

Area Under Curve (AUC)

The AUC, or Area Under the Curve, is a measure of a parameter's capacity to differentiate between two diagnostic classifications, such as normal and sick (Hajian-Tilaki, 2013). The AUC, which ranges from 0 to 1, measures the degree of discrimination provided by the parameter (Varoquaux & Colliot, 2023). A number close to 1 suggests a very reliable diagnostic outcome, indicating a great capacity to distinguish between the two groups.

RESULTS AND DISCUSSION

Data preprocessing

The section includes Importing the Libraries, Loading the Dataset, Printing the Head and Tail of the Dataset, Dataset Structure and Dimensions, Handling Missing Values, Dataset Information Overview, Descriptive Statistics of the Dataset, Crop Distribution, Crops Distribution Chart, and Outliers Detection.

Importing the Libraries

The Jupyter Notebook was set up with the necessary Python libraries, including Numpy, Pandas, Matplotlib, and Seaborn. Numpy is a powerful package that allows for efficient manipulation and concurrent operations on multi-dimensional arrays (Stančin & Jović, 2019). Panda is a data analysis and manipulation tool that is open-source and based on the Python programming language (Subasi, 2020). Matplotlib and Seaborn are extensively used software tools for data visualization. The program provides a user-friendly interface that simplifies the process of creating visually appealing and informative graphs. Seaborn, a data visualization software, is a subset of Matplotlib that offers a more limited set of functionalities (Pintor *et al*., 2019). Figure 2 depicts the process of importing the Numpy, Pandas, Matplotlib, and Seaborn Python libraries into the Jupyter Notebook.

```
import pandas as pd  # For data manipulation
import numpy as np  # For scientific computing
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import joblib
```

Figure 2: Importing Python Libraries

Loading the Dataset

The dataset was loaded into Jupyter Notebook using pd.read_csv. Figure 3 shows the loading of the dataset into Jupyter Notebook.

```
data = pd.read_csv('Crop_recommendation.csv')
data.head()
```

Figure 3: Loading the Dataset

The Head and Tail of the Dataset

The head () function displays the top records in the dataset depicted in Table 1. By default, python displays only the top 5 records while the tail () functions display the last five records of the dataset as illustrated in Table 2.

Table 1: First Five Rows of the Dataset

```
df.head()
```

	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

Table 2: Last Five Records of the Dataset

```
df.tail()
```

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
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

[2200 rows x 8 columns]

Dataset Structure and Dimensions

The shape () function displays the number of rows and columns of the dataset. Figure 4 shows there are 2200 number of rows and 8 number of columns.

.

```
data.shape
```

```
(2200, 8)
```

Figure 4: Number of Rows and Columns in the Dataset

Handling Missing Values

The presence of missing values was evaluated for each attribute in the dataset, and no instances of missing values were found. Figure 5 shows that there are no missing values (NaN) in any of the columns. Each column has 0 missing values, as indicated by the values in the right column (0 under each column name).

```
In [2]: df.isnull().sum()

Out[2]:
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

Figure 5: Checking for Missing Values in the dataset

Dataset Information Overview

The dataset was loaded into the Pandas data frame for easy analysis, model development, and prediction. Table 3 depicts the Pandas Data Frame comprising 2200 rows and 8 columns representing agricultural parameters. The columns include Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, Rainfall, and Label. The data types vary, with 3 integer columns (Nitrogen, Phosphorus, Potassium), 4 float columns (Temperature, Humidity, pH, Rainfall), and 1 object column (Label).

Table 3: Info of the Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Nitrogen     2200 non-null   int64
 1   phosphorus   2200 non-null   int64
 2   potassium    2200 non-null   int64
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object
```

Descriptive Statistics of Dataset

The *describe ()* method provides a concise summary of a dataset containing 2200 records and 7 columns, all of which include numerical values. Table 4 offers statistical data for several factors, including Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, and Rainfall. The dataset consists of statistical metrics, including the mean, standard deviation, minimum, maximum, and quartiles. These measurements provide valuable insights into the average values and variability of agricultural indicators, making it easier to analyze and identify any abnormal data points.

Table 4: Dataset Descriptive Statistics

	Nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

Crop Distribution

The frequency of rows matching each crop type in the "label" column is shown in Table 5. The value_counts() function computes the occurrence rate of each unique value in the given column. The dataset has 100 rows for each of the 22 crop varieties, including rice, maize, jute, cotton, coconut, papaya, orange, apple, muskmelon, watermelon, grapes, mango, banana, pomegranate, lentil, black gram, mungbean, mothbeans, pigeonpeas, kidney beans, chickpea, and coffee.

Table 3: Crop Distribution

Number of rows for each crop:			
rice	100	grapes	100
maize	100	mango	100
jute	100	banana	100
cotton	100	pomegranate	100
coconut	100	lentil	100
papaya	100	blackgram	100
orange	100	mungbean	100
apple	100	mothbeans	100
muskmelon	100	pigeonpeas	100
watermelon	100	kidneybeans	100
		chickpea	100
		coffee	100

Crops Distribution Chart

The pie chart was created using the Matplotlib library, illustrating the occurrence of each unique value in the "label" column of the Data Frame. The pie chart depicted in Figure 6 visually illustrates the distribution of different crop types in the dataset. Each segment of the chart represents a specific crop, and the size of each segment indicates the proportion of that crop in the dataset.

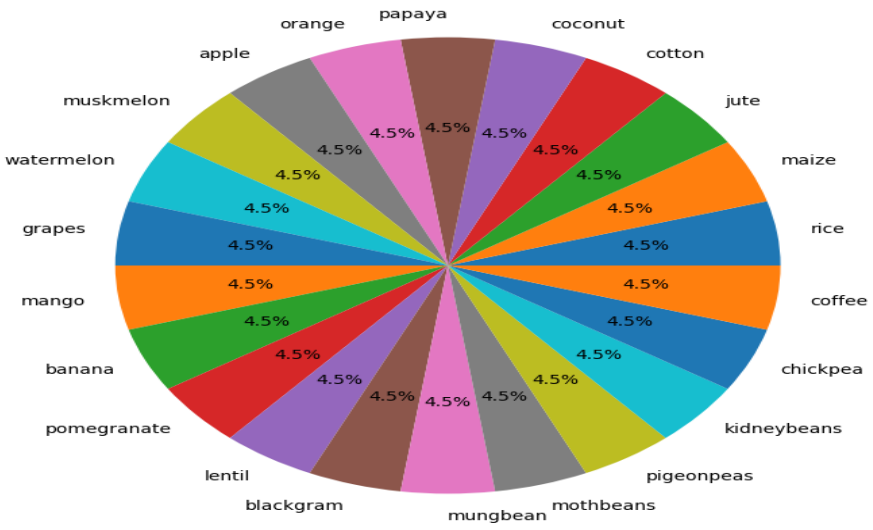

Figure 6: Crops Distribution Chart

Outliers Detection

The crop outlier distribution is shown using a boxplot illustrated in Figure 7. This allows for the comparison of quantitative data values across various groups. From the plots, it can be observed that there are many data points indicating outliers for all factors except Nitrogen. This is due to the different optimal soil conditions for the different plants.

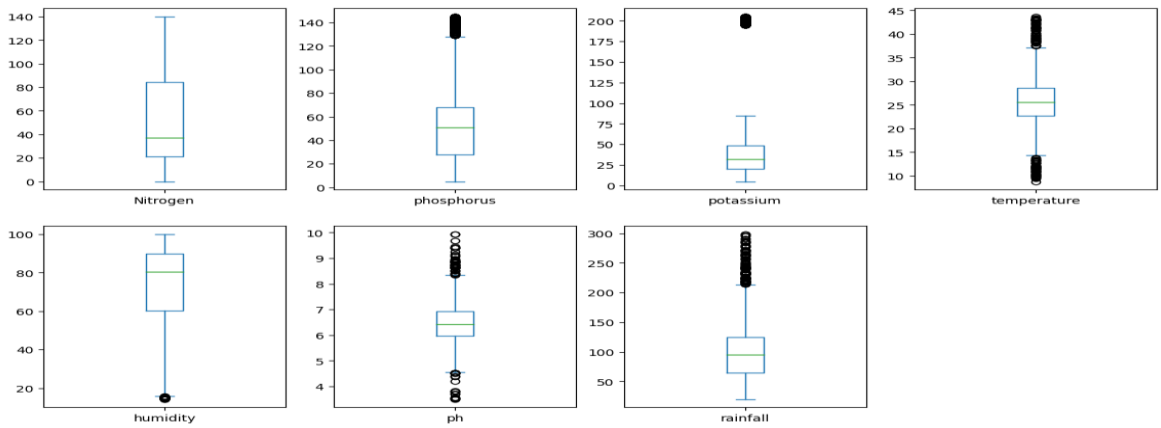

Figure 7: Outliers Detection of the Crop Distribution

Results

The results extracted from the findings of the data analysis are presented here. The results are presented using figures and tables.

Encoding of Target Variable

This is the conversion of letters in the dataset into figures. Figure 8 depicts how the Label Encoder object is created. This object was used to transform the categorical labels in the target variable y into numerical labels. The fit_transform method of the Label Encoder is applied to the target variable y. This method both fits the encoder to the unique labels in y and transforms the labels into numerical equivalents. The resulting encoded labels are stored in the variable y_encoded.

```
# # Encode the target variable into numeric values
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
#
```

Figure 8: Encode the target variable into numeric values

Getting the Correlation

This shows the correlation of each of the features to one another. Figure 9 calculates the correlation matrix (**corr**) for the features in the dataset (**data**) excluding the 'label' column. The **drop** method is used to exclude the 'label' column from the dataset.

```
corr= data.drop(['label'],axis=1).corr()
# sns.heatmap(corr,annot=True,cbar=True,cmap='coolwarm')

plt.figure(figsize = (10,10))
sns.heatmap(data.corr(),annot=True)
plt.show()
```

Figure 9: Getting the Correlation

The resulting data frame is then used to calculate the correlation matrix using the **corr()** method depicted in Figure 10.

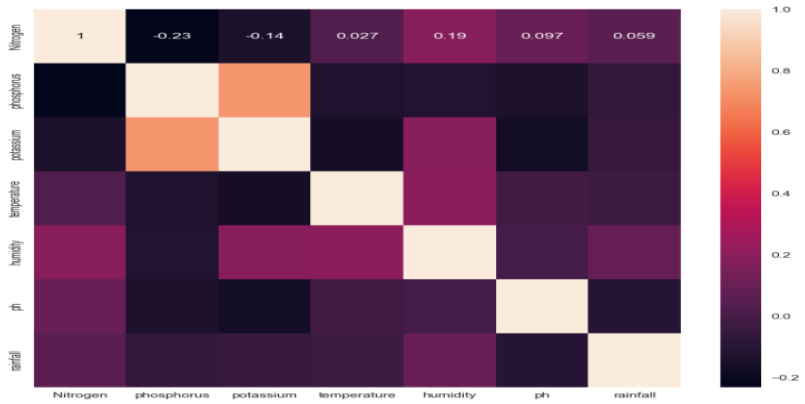

Figure 10: Correlation Matrix

Feature Scaling

The dataset was standardized using the Standard Scaler to improve model performance. Figure 11 illustrates the scaling and standardization of both the training and test data through the use of a standard scaler.

```
In [ ]: from sklearn.preprocessing import StandardScaler
        # scaler
        scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.fit_transform(X_test)
```

Figure 11: Feature scaling

Splitting the Dataset into two

The dataset is split into input variables (x) and the output variable (y). The **(x)** contains the features (independent variables) for each data point. The (**y)** containing the target label (dependent variable) corresponds to each data point illustrated in Figure 12.

```
In [ ]: from collections import Counter
        # Split the data into features (X) and the target label (y)
        X = data[['Nitrogen', 'phosphorus', 'potassium', 'temperature', 'humidity', 'ph', 'rainfall']]
        y = data['label']
```

Figure 12: Splitting the dataset into x and y

Discussion

Table 6 shows the comparative performance analysis of ensembling learning models. Four ensembling learning models. —namely, Stacking, Bagging, Voting, and One-Vs-Rest (OVR) were selected for this study. The models were trained and evaluated as individual classifiers for crop recommendation. The result of the models was analyzed and compared to derive insights to determine the model performance based on standard evaluation metrics such as accuracy, precision, recall, f1-score, and support.

Table 1: Accuracy Table of the Models

S/N	Model	Accuracy (%)	Precision	Recall	F1-score	Support
1	Bagging	99.32	0.99	1.00	1.00	141
2	OVR	91.81	1.00	0.96	0.98	141
3	Stacking	96.14	0.96	0.99	0.97	141
4	Voting	99.09	1.00	1.00	1.00	141

In comparing the performance of various machine learning models, it is evident that Bagging and Voting outperform other models in terms of accuracy. Bagging achieves the highest accuracy of 99.32% (Nti et al., 2023), closely followed by Voting with an accuracy of 99.09%. These two techniques, Bagging and Voting, consistently demonstrate exceptional accuracy across

different datasets, suggesting their robustness and reliability.

OVR (One vs Rest) approach achieves a relatively lower accuracy of 91.81% (Nti et al., 2023) compared to Bagging and Voting. However, it still demonstrates remarkable performance in efficiently differentiating between classes, as indicated by its high recall and F1-score values. Despite its lower accuracy, OVR remains a viable approach for tasks requiring class differentiation.

Stacking, although not reaching the same level of accuracy as Bagging or Voting, still achieves a respectable accuracy of 96.14%. It demonstrates balanced performance across various evaluation metrics, indicating its capability to make accurate predictions while minimizing false positives and false negatives.

Comparatively, other models such as Deep Reinforcement Learning (Khaki & Wang, 2019), traditional classifiers like KNN, NB, MLR, ANN, RF (Palanivel & Surianarayana, 2019), linear regression, LASSO, Light GBM, and ensemble methods like AdaBoost GB, and Stacked TBEL (Kalimuthu, Vaishnavi & Kishore, 2020), exhibit varying levels of performance, with accuracies ranging from 72.33% to 99.32%. These results highlight the importance of selecting appropriate machine learning techniques based on the specific requirements and characteristics of the dataset. The findings are displayed in a bar chart seen in Figure 13.

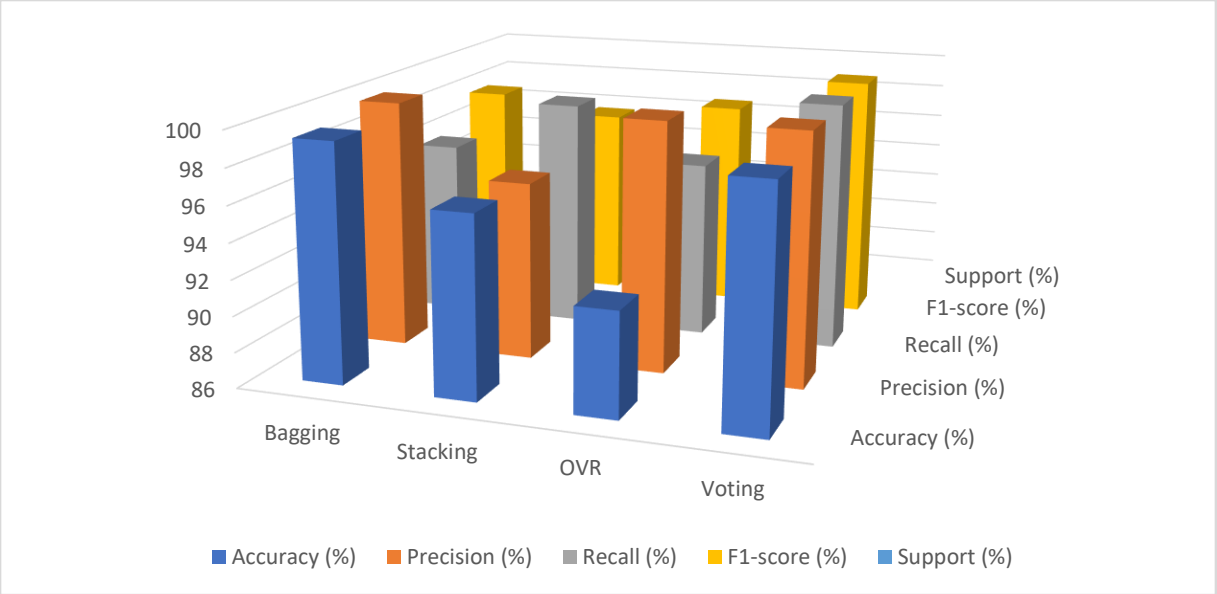

Figure 13: Comparative Analysis of All Models

CONCLUSION

The performed comparison analysis in this paper provides insights into the performance features of several ensemble learning models in the specific context of crop recommendation. By assessing Bagging, Voting, Stacking, and One-Vs-Rest (OVR), we have acquired a significant understanding of the merits and limitations of each method. Bagging and Voting are the most accurate and high-performing techniques. These models exhibit resilience in successfully forecasting crop recommendations, as seen by their high accuracy scores and excellent precision, recall, and F1-score values. Their capacity to combine forecasts from numerous foundational models significantly enhances their exceptional performance. Farmers and agricultural stakeholders in search of precise and dependable crop recommendation systems may find Bagging and Voting methods especially well-suited for their requirements. Nevertheless, Stacking and OVR emerge as feasible alternatives, showcasing robust accuracy and recall skills. Although Bagging and Voting may have somewhat higher accuracy compared to Stacking and OVR, the latter two techniques provide more subtle methods for ensemble learning. Stacking utilizes meta-modeling to merge predictions from many base models, whereas OVR adopts a one-vs-all technique to tackle challenges involving multi-class categorization. These models are more suitable in situations when achieving high levels of accuracy and completeness are crucial, or where the capacity to understand and explain the model's decisions is a top concern.

REFERENCES

Agrawal, N., Govil, H., & Kumar, T. (2024). Agricultural land suitability classification and crop suggestion using machine learning and spatial multicriteria decision analysis in semi-arid ecosystem. *Environment,*

Development and Sustainability, 1-38. https://doi.org/10.1007/s10668-023-04440-1

Akkem, Y., Biswas, S. K., & Varanasi, A. (2023). Smart farming using artificial intelligence: A review. *Engineering Applications of Artificial Intelligence*, *120*, 105899. https://doi.org/10.1016/j.engappai.2023.105899

Attri, I., Awasthi, L. K., Sharma, T. P., & Rathee, P. (2023). A review of deep learning techniques used in agriculture. Ecological Informatics, 102217. https://doi.org/10.1016/j.ecoinf.2023.102217

Avramopoulos, I., & Vasiloglou, N. (2023). On algorithmically boosting fixed-point computations. arXiv preprint arXiv:2304.04665. https://doi.org/10.48550/arXiv.2304.04665

Bach, J. (2020). When artificial intelligence becomes general enough to understand itself. Commentary on Pei Wang's paper "On defining artificial intelligence". Journal of Artificial General Intelligence, 11(2), 15-18. https://DOI: 10.2478/jagi-2020-0003

Behera, S., Menon, D., Shenoy, G. V., & Suresh, J. M. (2023, June). Suggestion of Appropriate Crops Based on Rainfall and Underground Water Analysis. In 2023 3rd International Conference on Intelligent Technologies (CONIT) (pp. 1-6). IEEE. doi: 0.1109/CONIT59222.2023.10205821

Biswas, S., Wardat, M., & Rajan, H. (2022, May). The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in the small, and the large. In Proceedings of the 44th International Conference on Software Engineering (pp. 2091-2103). https://doi.org/10.1145/3510003.3510057

Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140. https://doi.org/10.1007/BF00058655

Chakauya, R., Materechera, S. A., Jiri, O., Chakauya, E., & Machete, M. (2023). Climate change impacts on agriculture, adaptation and resilience: insights from local farmers in South-East Zimbabwe. In *Routledge Handbook of Climate Change Impacts on Indigenous Peoples and Local Communities* (pp. 273-286). Routledge. https://doi.org/10.4324/9781003356837

Chen, J., Zeb, A., Nanehkaran, Y. A., & Zhang, D. (2023). Stacking ensemble model of deep learning for plant disease recognition. Journal of Ambient Intelligence and Humanized Computing, 14(9), 12359-12372. https://doi.org/10.1007/s12652-022-04334-6

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree-boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). https://doi.org/10.1145/2939672.2939785

Davis, E. (2015). Ethical guidelines for a superintelligence. Artificial Intelligence, 220, 121-124. https://doi.org/10.1016/j.artint.2014.12.003

Demir, S., & Sahin, E. K. (2023). Predicting occurrence of liquefaction-induced lateral spreading using gradient boosting algorithms integrated with particle swarm optimization: PSO-XGBoost, PSO-LightGBM, and PSO-CatBoost. Acta Geotechnica, 18(6), 3403-3419. https://doi.org/10.1007/s11440-022-01777-1

De-Zarzà, I., de Curtò, J., Hernández-Orallo, E., & Calafate, C. T. (2023). Cascading and Ensemble Techniques in Deep Learning. Electronics, 12(15), 3354. https://doi.org/10.3390/electronics12153354

Emami, S., & Martínez-Muñoz, G. (2023). Sequential Training of Neural Networks with Gradient Boosting. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3271515

Everitt, T., Goertzel, B., & Potapov, A. (2017). Artificial general intelligence. Lecture Notes in Artificial Intelligence. Heidelberg: Springer. https://doi.org/10.1007/978-3-319-63703-7

Falcon, W. P., Naylor, R. L., & Shankar, N. D. (2022). Rethinking global food demand for 2050. *Population and Development Review*, *48*(4), 921-957. https://doi.org/10.1111/padr.12508

Flasiński, M. (2016). Introduction to artificial intelligence. Springer. https://doi: 10.1007/978-3-319-40022-8

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, *115*, 105151. https://doi.org/10.1016/j.engappai.2022.105151

González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical

perspectives and opportunities. Information Fusion, 64, 205-237. https://doi.org/10.1016/j.inffus.2020.07.007

Gul, N., Mashwani, W. K., Aamir, M., Aldahmani, S., & Khan, Z. (2023). Optimal model selection for k-nearest neighbours ensemble via sub-bagging and sub-sampling with feature weighting. Alexandria Engineering Journal, 72, 157-168. https://doi.org/10.1016/j.aej.2023.03.075

Hajian-Tilaki K. Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. Caspian J Intern Med. 2013 Spring;4(2):627-35. PMID: 24009950; PMCID: PMC3755824.

Han, R., Yoon, H., Kim, G., Lee, H., & Lee, Y. (2023). Revolutionizing Medicinal Chemistry: The Application of Artificial Intelligence (AI) in Early Drug Discovery. Pharmaceuticals (Basel, Switzerland), 16(9), 1259. https://doi.org/10.3390/ph16091259.

Hutter, M. (2012). One decade of universal artificial intelligence. In Theoretical foundations of artificial general intelligence (pp. 67-88). Paris: Atlantis Press. https://doi.org/10.2991/978-94-91216-62-6_5

Jia, J., Liang, W., & Liang, Y. (2023). A Review of Hybrid and Ensemble in Deep Learning for Natural Language Processing. arXiv preprint arXiv:2312.05589. https://doi.org/10.48550/arXiv.2312.05589

Jitpakdeebodin, W., & Sinapiromsaran, K. (2023, March). Random forest algorithm using quartile-pattern bootstrapping for a class imbalanced problem. In Proceedings of the 2023 5th International Conference on Image, Video and Signal Processing (pp. 191-196). https://doi.org/10.1145/3591156.3591184

Kalimuthu, M., Vaishnavi, P., & Kishore, M. (2020, August). Crop prediction using machine learning. In *2020 third international conference on smart systems and inventive technology (ICSSIT)* (pp. 926-932). IEEE. https://doi.org/10.1109/ICSSIT48917.2020.9214190

Khaki, S., & Wang, L. (2019). Crop yield prediction using deep neural networks. *Frontiers in plant science*, *10*, 452963. https://doi.org/10.3389/fpls.2019.00621

Kom, Z., Nethengwe, N. S., Mpandeli, N. S., & Chikoore, H. (2022). Determinants of small-scale farmers' choice and adaptive strategies in response to climatic shocks in Vhembe District, South Africa. *GeoJournal*, *87*(2), 677-700. https://doi.org/10.1007/s10708-020-10272-7

Liang, W., & Liu, Y. (2023). Rating Crop Insurance Contracts with Model Stacking of Gaussian Processes.

Limpo, S. Y., Fahmid, I. M., Fattah, A., Rauf, A. W., Surmaini, E., Muslimin, ... & Andri, K. B. (2022). Integrating indigenous and scientific knowledge for decision making of rice farming in South Sulawesi, Indonesia. *Sustainability*, *14*(5), 2952. https://doi.org/10.3390/su14052952

Maganathan, T., Senthilkumar, S., & Balakrishnan, V. (2020, November). Machine learning and data analytics for environmental science: a review, prospects and challenges. In *IOP conference series: materials science and engineering* (Vol. 955, No. 1, p. 012107). IOP Publishing. https://**doi.org**/10.1088/1757-899X/955/1/012107

Malek, N. H. A., Yaacob, W. F. W., Wah, Y. B., Nasir, S. A. M., Shaadan, N., & Indratno, S. W. (2023). Comparison of ensemble hybrid sampling with bagging and boosting machine learning approach for imbalanced data. Indones. J. Elec. Eng. Comput. Sci, 29, 598-608. https://doi.org/10.11591/ijeecs.v29.i1.pp598-608

Miller, T., Lewita, K., Krzemińska, A., Kozlovska, P., Jawor, M., Cembrowska-Lech, D., & Kisiel, A. (2023). Boosting modern society: advancements and applications of the adaboost algorithm in diverse domains. *Scientific Collection «InterConf»*, (152), 549–555. Retrieved from https://archive.interconf.center/index.php/conference-proceeding/article/view/3214

Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences*, *35*(2), 757-774. https://doi.org/10.1016/j.jksuci.2023.01.014

Nalluri, M., Pentela, M., & Eluri, N. R. (2020). A Scalable Tree Boosting System: XG Boost. Int. J. Res. Stud. Sci. Eng. Technol, 7, 36-51. https://doi.org/10.22259/2349-476X.0712005

Nosrati, V., & Rahmani, M. (2023). HMDE-FS: A homogeneous distributed ensemble feature selection framework based on resampling with/without replacement. Concurrency and Computation: Practice and Experience, 35(7), e7613. https://doi.org/10.1002/cpe.7613

Nti, I. K., Zaman, A., Nyarko-Boateng, O., Adekoya, A. F., & Keyeremeh, F. (2023). A predictive analytics model for crop suitability and productivity with tree-based

ensemble learning. Decision Analytics Journal, 8, 100311. https://doi.org/10.1016/j.dajour.2023.100311

Palanivel, K., & Surianarayanan, C. (2019). An Approach for Prediction of Crop Yield Using Machine Learning and Big Data Techniques (2019). International Journal of Computer Engineering and Technology 10(3), pp. 110-118, 2019, Available at SSRN: https://ssrn.com/abstract=3555087

Petropoulos, F., & Siemsen, E. (2023). Forecast selection and representativeness. Management Science, 69(5), 2672-2690. https://doi.org/10.1287/mnsc.2022.4485

Pintor, M., Demetrio, L., Sotgiu, A., Melis, M., Demontis, A., & Biggio, B. (2019). secml: A Python Library for Secure and Explainable Machine Learning. arXiv preprint arXiv:1912.10013. https://doi.org/10.1016/j.softx.2022.101095

Rana, M., Chandorkar, P., Dsouza, A., & Kazi, N. (2015). Breast Cancer Diagnosis and Recurrence Prediction Using Machine Learning Techniques. *International Journal of Research in Engineering and Technology*, *4*(4), 372–376. eISSN: 2319-1163 | pISSN: 2321-7308

Redhu, N. S., Thakur, Z., Yashveer, S., & Mor, P. (2022). Artificial intelligence: a way forward for agricultural sciences. In *Bioinformatics in Agriculture* (pp. 641-668). Academic Press. https://doi.org/10.1016/B978-0-323-89778-5.00007-6

Ricciardi, V., Ramankutty, N., Mehrabi, Z., Jarvis, L., & Chookolingo, B. (2018). An open-access dataset of crop production by farm size from agricultural censuses and surveys. Data in brief, 19, 1970-1988. https://doi.org/10.1016/j.dib.2018.06.057

Sajitha, P., Andrushia, A. D., Anand, N., & Naser, M. Z. (2024). A Review on Machine Learning and Deep Learning Image-based Plant Disease Classification for Industrial Farming Systems. *Journal of Industrial Information Integration*, 100572. https://doi.org/10.1016/j.jii.2024.100572

Seireg, H. R., Omar, Y. M., & Elmahalawy, A. (2023, October). Multi-Agent System Based on Stacking Technique for Rice Yield Prediction. In 2023 3rd International Conference on Electronic Engineering (ICEEM) (pp. 1-6). IEEE. https://doi.org/10.1109/ICEEM58740.2023.10319559

Serraj, R., Krishnan, L., & Pingali, P. (2019). Agriculture and food systems to 2050: a synthesis. *Agriculture & Food Systems to*, *2050*, 3-45. Retrieved from https://www.worldscientific.com/doi/pdf/10.1142/11212#page=20

Shaikh, T. A., Rasool, T., & Lone, F. R. (2022). Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming. *Computers and Electronics in Agriculture*, *198*, 107119. https://doi.org/10.1016/j.compag.2022.107119

Shams, M. Y., Gamel, S. A., & Talaat, F. M. (2024). Enhancing crop recommendation systems with explainable artificial intelligence: a study on agricultural decision-making. *Neural Computing and Applications*, 1-20. https://doi.org/10.1007/s00521-023-09391-2

Stančin, I., & Jović, A. (2019). An overview and comparison of free Python libraries for data mining and big data analysis. 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 977–982. https://doi.org/10.23919/MIPRO.2019.8757088

Subasi, A. (2020). Practical Machine Learning for Data Analysis. Academic press. http://dx.doi.org/10.1016/B978-0-12-821379-7.00001-1

Sun, F., Meng, X., Zhang, Y., Wang, Y., Jiang, H., & Liu, P. (2023). Agricultural Product Price Forecasting Methods: A Review. Agriculture, 13(9), 1671. https://doi.org/10.3390/agriculture13091671

Thotad, P. N., Bharamagoudar, G. R., & Kallur, S. S. (2023). Boosting-based machine learning approaches for diabetes prediction using Indian demographic and health survey-2021 data. https://doi.org/10.21203/rs.3.rs-2784266/v1

Tripathi, A. D., Mishra, R., Maurya, K. K., Singh, R. B., & Wilson, D. W. (2019). Estimates for world population and global food availability for global health. In *The role of functional food security in global health* (pp. 3-24). Academic Press. https://doi.org/10.1016/B978-0-12-813148-0.00001-3

Varoquaux, G., & Colliot, O. (2023). Evaluating machine learning models and their diagnostic value. Machine Learning for Brain Disorders, 601-630. https://doi.org/10.1007/978-1-0716-3195-9_20

Yang, L., Yu, X., Zhang, S., Zhang, H., Xu, S., Long, H., & Zhu, Y. (2023). Stacking-based and improved convolutional neural network: a new approach in rice leaf disease identification. Frontiers in Plant Science, 14, 1165940. https://doi.org/10.3389/fpls.2023.1165940

Zhang, J., Liu, J., Chen, Y., Feng, X., & Sun, Z. (2021). Knowledge mapping of machine learning approaches applied in agricultural management—a scientometric review with citespace. *Sustainability*, *13*(14), 7662. https://doi.org/10.3390/su13147662

Zounemat-Kermani, M., Batelaan, O., Fadaee, M., & Hinkelmann, R. (2021). Ensemble machine learning paradigms in hydrology: A review. *Journal of Hydrology*, *598*, 126266. https://doi.org/10.1016/j.jhydrol.2021.126266