



Performance Evaluation of Machine Learning Models for Crop Yield Prediction

Muhammad Umar Abdullahi^{1*}, Gilbert I.O. Aimufua², Morufu Olalere², Kene Tochukwu Anyachebelu²,
Tahir Abdulhakim²

¹Department of Computer Science, Federal University of Technology, Owerri, Nigeria

²Department of Computer Science, Nasarawa State University, Keffi, Nigeria

Corresponding Author's Mail: umarmuhammadmuhammad@nsuk.edu.ng

Received on: January, 2024

Revised and Accepted on: March 2024

Published on: March 2024

ABSTRACT

Agriculture, a fundamental pillar of worldwide sustenance, greatly benefits from precise yield projections, which provide effective allocation of resources and well-informed decision-making. This work focuses on the crucial task of predicting agricultural yields by conducting a thorough comparative examination of several machine-learning models. The examined models include Linear Regression, Random Forest, Extreme Gradient Boost (XGBoost), K-Nearest Neighbors (KNN), Decision Tree, and Bagging Regressor. The results demonstrate subtle variations in performance, with Linear Regression highlighting constraints in its ability to make accurate predictions. Ensemble approaches, namely: Random Forest and XGBoost, demonstrate remarkable accuracy, achieving almost 97% and R^2 ratings. This highlights their ability to effectively capture complex agricultural patterns. The findings of this research provide valuable suggestions for professionals in agriculture and machine learning, making it easier to choose reliable models for predicting crop yields. It is also recommended to optimize Random Forest and XGBoost for accurate production predictions in practical agricultural scenarios. Future studies may focus on using sophisticated optimization approaches and incorporating specialized domain knowledge to enhance the precision of agricultural production prediction.

Keywords: Performance Evaluation, Machine Learning Models, Crop Yield Prediction, Evaluation Metrics and Model Assessment

1.0 INTRODUCTION

Agriculture serves as the backbone of economies worldwide, providing sustenance and livelihoods for a significant portion of the global population. Predicting crop yields accurately is essential for optimizing agricultural practices, resource allocation, and ensuring food security (Al-Adhaileh & Aldhyani, 2022; Nti, Zaman, Nyarko-Boateng, Adekoya & Keyeremeh, 2023).

Traditional methods of crop yield prediction, relying on historical data and simplistic statistical models, often struggle to capture the complexities inherent in agricultural systems (Engen *et al.*, 2021; Sun *et al.*, 2023). In recent years, the integration of machine learning (ML) models has emerged as a promising avenue for improving the accuracy and efficiency of crop yield predictions (Araújo, Peres, Ramalho, Lidon & Barata, 2023; Droutsas, Challinor, Deva & Wang, 2022).

Machine learning, a subset of artificial intelligence, offers a data-driven approach that can analyze vast and diverse datasets to discern intricate patterns and relationships (Aldoseri, Al-Khalifa, & Hamouda, 2023). The application of ML in crop yield prediction involves leveraging a variety of factors, including but not limited to remote sensing data, weather patterns, soil

characteristics, and historical yield records (Han, Yoon, Kim, Lee & Lee, 2023). The capability of ML models to adapt to changing conditions and learn from dynamic datasets makes them well-suited for addressing the challenges posed by the variability in agricultural environments (Elahi, Afolaranmi, Martinez-Lastra & Perez Garcia, 2023).

This study focuses on the performance evaluation of several machine learning models in the context of crop yield prediction. The selected models for evaluation include Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors (KNN), Decision Tree, and Bagging Regressor. Each of these models represents a distinct approach to machine learning, with unique strengths and characteristics that may impact their effectiveness in predicting crop yields (Dhoni, 2023).

Linear Regression provides a baseline for understanding the linear relationships between input features and crop yields (Sarr & Sultan, 2023). Random Forest and Gradient Boosting are ensemble methods that combine multiple decision trees to enhance predictive accuracy (Omer & Shareef, 2022). XGBoost, an optimized implementation of gradient boosting, further refines the

performance of ensemble models (Chen & Guestrin, 2016; Nalluri, Pentela & Eluri, 2020; Ahn, Kim & Kim, 2023). KNN operates on the principle of similarity, considering the proximity of data points in a multidimensional space. Decision Tree models are intuitive and offer interpretability (Cañete-Sifuentes, Monroy & Medina-Pérez, 2021), while Bagging Regressor combines the strengths of multiple models for improved stability and accuracy (Plaia, Buscemi, Fürkranz & Mencía, 2022; Costa, V. G., & Pedreira, 2023).

The significance of this study lies in its potential to offer insights into the comparative performance of these diverse machine-learning models for crop yield prediction. By systematically evaluating and contrasting their accuracy, robustness, and computational efficiency, this research aims to guide practitioners, researchers, and

policymakers in selecting the most suitable model for their specific agricultural contexts. Additionally, the study contributes to advancing the broader understanding of the role of machine learning in addressing the challenges faced by modern agriculture, such as: climate variability, resource optimization, and sustainable food production.

2.0 METHODOLOGY

The methodology for this paper is known as Machine Learning Pipeline or Data Science Workflow (Biswas, Wardat & Rajan, 2022). The methodology has eight stages. As shown in Figure 1. which include: dataset collection, data exploration, and pre-processing, splitting dataset, training the models, performance evaluation, performance comparison, conclusion (Biswas, Wardat & Rajan, 2022).

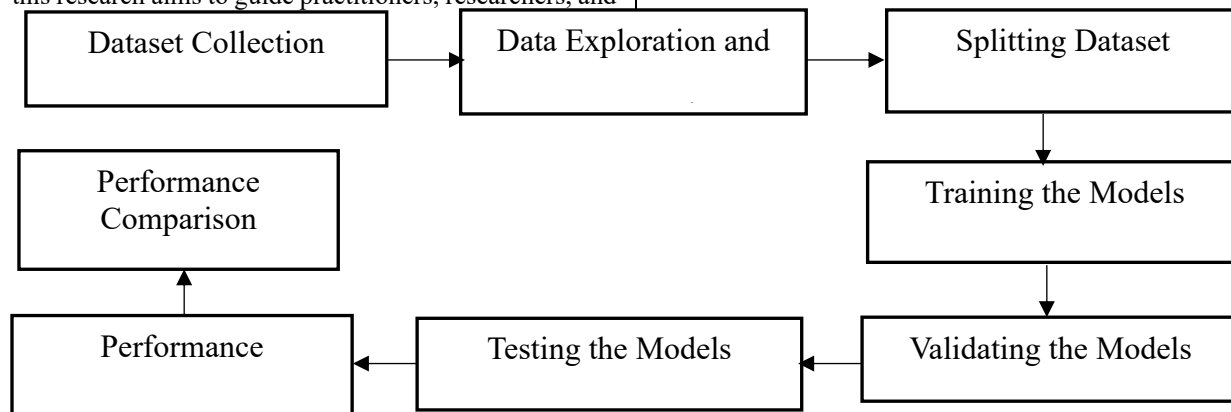


Figure 1: Data Science Workflow
Source: Author Based on (Rana *et al.*, 2015)

Figure 1 shown in this study was modified from the original work of Rana *et al.* (2015) to align with the research approach utilized in this report. The attainment of this target was achieved by the systematic implementation of a well-structured series of procedures.

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 machine learning models were utilized, specifically Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors (KNN), Decision Tree, and Bagging Regressor. 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 breast cancer, therefore adding vital insights to the area of medical diagnostics.

2.1 Source of Dataset

The dataset was obtained from <https://www.kaggle.com/datasets/patelris/crop-yield-prediction-dataset>. The dataset encompasses data from Food and Agriculture Organization and World Data Bank. The data size was 28242 records and six predictors. The target variable consists of six classes representing different crops (i.e., 'Maize', 'Potatoes', 'Rice', 'Sorghum' and 'Wheat') each with one hundred (100) samples.

2.2 Performance Metrics for Classification

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

2.2.1 Accuracy

The efficacy of a model is assessed by the proportion of accurate predictions produced across all sorts of forecasts (Ricciardi, Ramankutty, Mehrabi, Jarvis & Chookolingo, 2018). The evaluation process involves assessing the accuracy of classification by comparing the count of correctly categorized instances to the overall count of occurrences (Petropoulos & Siemsen, 2023). The measure of accuracy is particularly valuable in cases when the distribution of classes in the target variable is uniformly spread throughout the dataset. This is expressed in Equation 1.

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

2.2.3 Sensitivity or Recall

Sensitivity is a numerical metric used to calculate the proportion of correctly identified positive situations that were incorrectly labeled as negative by the model. It is sometimes denoted as recall or true positive rate (Hutter, 2012). Mathematically, it is defined as the ratio of the number of true positive (TP) occurrences to the sum of

true positive and false negative (FN) cases. It is mathematically expressed as:

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

2.2.4 Specificity

Specificity, known as the genuine negative rate, holds relevance within the software defect domain (Everitt, Goertzel & Potapov, 2017). Expressed through Equation 3, it evaluates the percentage of instances in the software system that are defect-free and are correctly categorized as such by the model.

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

2.2.5 Detection Rate

The detection rate refers to the proportion of the entire sample in which events were accurately identified (Flasiński, 2016). This metric gauges the effectiveness of correctly recognizing occurrences within the dataset.

2.2.6 F1 score rate

The F1 score represents the computed weighted average of both precision and recall (Bach, 2020). As such, this score takes into account the balance between false positives and false negatives.

$$F1 \text{ Score} = 2 * \frac{precision*recall}{precision+recall} \quad \dots (4)$$

2.2.7 Precision

Precision is a metric that measures the accuracy of positive predictions made by a model (Davis, 2015). It is defined as the ratio of correctly predicted positive samples to the total number of samples predicted as positive.

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

2.2.8 Area Under Curve (AUC):

The AUC (Area Under the Curve) serves as a gauge of a parameter's ability to distinguish between two diagnostic classes, such as normal and diseased (Hajian-Tilaki, 2013). Ranging from 0 to 1, the AUC quantifies the discriminatory power of the parameter (Varoquaux & Colliot, 2023). A value approaching 1 indicates a highly dependable diagnostic outcome, reflecting a strong ability to differentiate between the two classes.

3.0 RESULTS AND DISCUSSION

This section evaluates the proposed model's numerical experimental performance using the PROMISE dataset, varying sample and feature counts. The outcomes are

displayed in tables and graphs for a comprehensive presentation.

3.1 Data preprocessing

3.1.1 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

3.1.2 Loading the Dataset

The dataset was imported into the Jupyter Notebook environment using the `pd.read_csv` function. Figure 3 depicts the process of importing the dataset into Jupyter Notebook.

```
y_data = pd.read_csv("yield_df.csv")
y_data = y_data.drop(columns=y_data.filter(regex='Unnamed').columns)
y_data.head()
```

Figure 3: Loading the Dataset

3.1.3 Dataset Structure and Dimensions

The `shape()` function is employed to get and display the dimensions of a dataset, namely the count of rows and columns. Figure 4 depicts the presence of 28242 rows and 5 columns.

```
y_data.shape
```

```
(28242, 5)
```

Figure 4: Number of Rows and Columns in the Dataset

3.1.4 Duplicate Entries in the Dataset

The `duplicate()` function in a data frame returns a series of true and false values indicating which rows are duplicates. Table 1 shows that there are no duplicates in the dataset.

Table 1: Number of Duplicates

```
y_data.isnull().sum()
```

```
crop          0
yield         0
rainfall      0
pesticide     0
avg_temp      0
dtype: int64
```

3.1.5 Dataset Information Overview

The `info()` function is utilized to present the number of columns, their respective labels, data kinds, and the count of non-null cells within each column. According to the data presented in Table 2, there is a lack of empty values within the dataset. However, it is necessary to convert the diagnostic column from a string format to numerical values to provide more effective analysis utilizing machine learning methodologies.

Table 2: Info of the Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28242 entries, 0 to 28241
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   crop            28242 non-null  object
1   yield           28242 non-null  int64
2   rainfall        28242 non-null  float64
3   pesticide       28242 non-null  float64
4   avg_temp        28242 non-null  float64
```

3.1.6 Descriptive Statistics of Dataset

The `describe()` function offers a succinct overview of a dataset including 28242 items and 5 columns, all of which consist of integer values. Table 3 presents statistical data on many parameters, such as yield, rainfall, pesticide, and average temperature. The dataset comprises statistical measurements such as the mean, standard deviation, minimum, maximum, and quartiles. These metrics provide vital insights into the mean values and variability of agricultural indicators, facilitating the analysis and detection of any anomalous data points.

Table 3: Dataset Descriptive Statistics

	Yield	Rainfall	Pesticide	Avg temp
Count	28242.000000	28242.000000	28242.000000	28242.000000
Mean	77053.332094	1149.05598	37076.909344	20.542627
Std	84956.612897	709.81215	59958.784665	6.312051
Min	50.000000	51.00000	0.040000	1.300000
25%	19919.250000	593.00000	1702.000000	16.702500
50%	38295.000000	1083.00000	17529.440000	21.510000
75%	104676.750000	1668.00000	48687.880000	26.000000
Max	501412.000000	3240.00000	367778.000000	30.650000

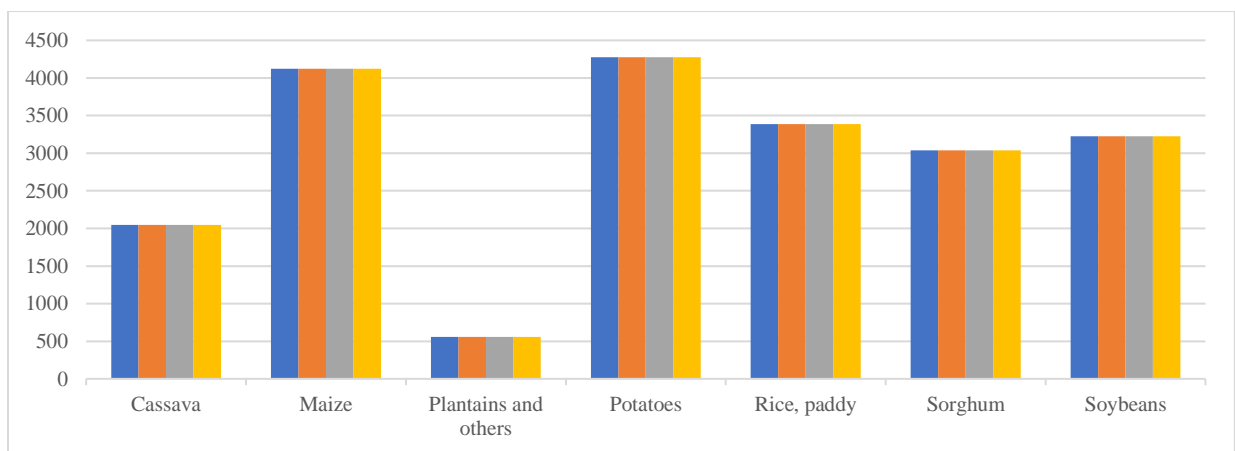
3.1.7 Grouping and Counting Instances by Crop in the Dataset

This grouping and counting operation aids in understanding the dataset's distribution and provides insights into each crop's data availability and representation. Table 4 depicts the count of occurrences for each crop category in the 'y_data' dataset across features such as yield, rainfall, pesticide, and average temperature.

Table 4: Grouping and Counting Instances by Crop in the Dataset

Crops	Yield	Rainfall	Pesticide	Avg temp
Cassava	2045	2045	2045	2045
Maize	4121	4121	4121	4121
Plantains and others	556	556	556	556
Potatoes	4276	4276	4276	4276
Rice, paddy	3388	3388	3388	3388
Sorghum	3039	3039	3039	3039
Soybeans	3223	3223	3223	3223
Sweet potatoes	2890	2890	2890	2890
Wheat	3857	3857	3857	3857
Yams	847	847	847	847

The Seaborn library in Python was used to create a count plot, displaying the distribution of occurrences for each unique crop in the 'y_data' dataset along the y-axis. This visualization in Figure 5 helps assess the dataset's balance and the frequency of each crop category.

**Figure 5: Distribution of Each Crop in the 'y_data' Dataset Along the y-Axis**

3.2 Results

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

3.2.1 Data Encoding

This involves the transformation of alphabetic characters in the dataset into numerical values (Elmagarmid, Ipeirotis, & Verykios, 2006; Webb, Holyoak & Lu, 2023). Figure 6 illustrates the process of creating the Label Encoder object. This object was used to convert the category labels in the target variable *y* into numerical labels. The `fit_transform` method of the Label Encoder is used on the target variable *y*. This approach simultaneously adapts the encoder to the distinct labels in *y* and converts the labels into numerical representations. The encoded labels obtained are kept in the variable `y_encoded`.

```
from sklearn.preprocessing import LabelEncoder
categorical_columns = datacorr.select_dtypes(include=['object']).columns.tolist()
yield_label_encoder = LabelEncoder()
for column in categorical_columns:
    datacorr[column] = yield_label_encoder.fit_transform(datacorr[column])
```

	crop	yield	rainfall	pesticide	avg_temp
0	Maize	36613	1485.0	121.00	16.37
1	Potatoes	66667	1485.0	121.00	16.37
2	Rice, paddy	23333	1485.0	121.00	16.37
3	Sorghum	12500	1485.0	121.00	16.37
4	Soybeans	7000	1485.0	121.00	16.37
...
28237	Rice, paddy	22581	657.0	2550.07	19.76
28238	Sorghum	3066	657.0	2550.07	19.76
28239	Soybeans	13142	657.0	2550.07	19.76
28240	Sweet potatoes	22222	657.0	2550.07	19.76
28241	Wheat	22888	657.0	2550.07	19.76

Figure 6: Data Encoding

3.2.2 Getting the Correlation

Table 5 indicates that among the variable's 'pesticide', 'rainfall', and 'avg_temp', 'pesticide' has the highest positive correlation with 'yield'. The correlation coefficient is approximately 0.0641, suggesting a weak positive correlation. This means that as the 'pesticide' variable increases, there is a slight tendency for the 'yield' variable to increase as well.

Table 5: Getting the Correlation

	Yield	Rainfall	Pesticide	Avg temp
Yield	1.000000	0.000962	0.064085	-0.114777
Rainfall	0.000962	1.000000	0.180984	0.313040
Pesticide	0.064085	0.180984	1.000000	0.030946
Avg temp	-0.114777	0.313040	0.030946	1.000000

3.2.3 Dropping Features

To enhance the classification accuracy of the model as seen in Figure 7, it was concluded that specific modifications should be implemented. In particular, one specific feature, namely yield, was selected for exclusion.

```
[ ] X, y = datacorr.drop(labels='yield', axis=1), datacorr['yield']
```

Figure 7: Dropping Features

3.2.4 Feature Scaling

Feature scaling was applied using Standard Scaler to standardize the dataset to optimize the performance of the models. Figure 8 shows the scaling and standardizing of the training, validation, and test data using a standard scaler.

```
In [21]: # feature scaling
from sklearn.preprocessing import StandardScaler
x_train=StandardScaler().fit_transform(x_train)
x_test=StandardScaler().fit_transform(x_test)
x_val=StandardScaler().fit_transform(x_val)
```

Figure 8: Feature scaling

3.2.5 Splitting the Dataset into two

The dataset has been partitioned into two distinct subsets, namely the test set and the training set. The data partitioning into these sets is seen in Figure 9. The training set comprises 80% of the dataset, and the test set encompasses 20% of the dataset.

```
In [16]: # split the dataset into dependent(X) and Independent(Y) database
x=df.iloc[:,2:28].values
y=df.iloc[:,1].values
```

Figure 9: Splitting the dataset into x and y

3.2.6 Modeling with the Selected Algorithm

The process involves the integration of the dataset into the algorithms to facilitate the training, validation, and testing of the models. Figure 10 illustrates the utilization of Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors (KNN), Decision Tree, and Bagging Regressor Algorithms to fit the training and validation data for training and validation purposes.

```
models = [
    ('Linear Regression', LinearRegression()),
    ('Random Forest', RandomForestRegressor(random_state=42)),
    ('Gradient Boost', GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)),
    ('XGBoost', XGBRegressor(random_state=42)),
    ('KNN', KNeighborsRegressor(n_neighbors=5)),
    ('Decision Tree', DecisionTreeRegressor(random_state=42)),
    ('Bagging Regressor', BaggingRegressor(n_estimators=150, random_state=42))
]
```

Figure 10: Modeling with the Selected Algorithm

4.0 DISCUSSION

The comparative analysis of machine learning models for the task of crop yield prediction offers valuable insights into their performance. Linear regression shows limited predictive power with an accuracy of 7.93% and an R2 score of 7.93%, indicating challenges in capturing the underlying patterns in the data. In contrast, ensemble methods, including Random Forest, XGBoost, and Bagging Regressor, demonstrate exceptional predictive capabilities with accuracy scores of around 97% and high R2 scores. Random Forest and XGBoost particularly stand out, showcasing their ability to capture complex relationships in the dataset. Gradient Boosting performs well but lags slightly behind the ensemble methods. K-

Nearest Neighbors (KNN) struggles with a significantly lower accuracy of 36.65%, suggesting challenges in its ability to generalize to the underlying patterns. Decision Tree performs admirably, though slightly below the performance of ensemble methods. Considering the outstanding results of Random Forest, XGBoost, and Bagging Regressor depicted in Table 6, further refinement and optimization of these ensemble models are recommended. Based on the current analysis, Random Forest and XGBoost emerge as the most promising models, demonstrating superior accuracy and robust performance. The final choice between them may depend on factors such as interpretability, computational efficiency, and specific requirements of the task.

Table 6: Accuracy Table of the Models

S/N	Model	Accuracy	MSE	R2 score
1	Linear Regression	0.079307	6824308871.305003	0.079307
2	Random Forest	0.969795	223887229.800693	0.969795
3	Gradient Boost	0.820810	1328181058.235154	0.820810
4	XGBoost	0.961757	283459459.381772	0.961757
5	KNN	0.366483	4695723314.925967	0.366483
6	Decision Tree	0.949647	373221220.941278	0.949647
7	Bagging Regressor	0.969561	225618851.570496	0.969561



5.0 CONCLUSION

This paper presented a comprehensive comparative analysis of machine learning models for the task of predicting crop yield. The results underscore the diverse performance characteristics across various models, shedding light on their strengths and limitations. Linear regression, while serving as a baseline, exhibited limited predictive power, emphasizing the need for more sophisticated approaches. Ensemble methods, particularly Random Forest, XGBoost, and Bagging Regressor, emerged as standout performers with exceptional accuracy and R2 scores, showcasing their ability to capture intricate relationships within the agricultural dataset. Notably, Random Forest and XGBoost demonstrated superior predictive capabilities, making them prime candidates for further optimization. This study contributes valuable insights to the field of agricultural prediction models, offering practitioners and researchers guidance in selecting robust models for crop yield estimation. Future research endeavors could explore additional optimization techniques, feature engineering, and the integration of domain-specific knowledge to enhance the predictive accuracy and interpretability of machine learning models in the agricultural domain.

Competing interests: The authors declare that they have no conflict of interest.

REFERENCE

- Ahn, J. M., Kim, J., & Kim, K. (2023). Ensemble Machine Learning of Gradient Boosting (XGBoost, LightGBM, CatBoost) and Attention-Based CNN-LSTM for Harmful Algal Blooms Forecasting. *Toxins*, 15(10), 608. <https://doi.org/10.3390/toxins15100608>.
- Al-Adhaileh, M. H., & Aldhyani, T. H. (2022). Artificial intelligence framework for modeling and predicting crop yield to enhance food security in Saudi Arabia. *PeerJ Computer Science*, 8, e1104. <https://doi.org/10.7717/peerj-cs.1104>.
- Aldoseri, A., Al-Khalifa, K. N., & Hamouda, A. M. (2023). Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges. *Applied Sciences*, 13(12), 7082. <https://doi.org/10.3390/app13127082>
- Araújo, S. O., Peres, R. S., Ramalho, J. C., Lidon, F., & Barata, J. (2023). Machine Learning Applications in Agriculture: Current Trends, Challenges, and Future Perspectives. *Agronomy*, 13(12), 2976. <https://doi.org/10.3390/app13127082>.
- 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. DOI: 10.2478/jagi-2020-0003.
- 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 in the large. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 2091-2103). <https://doi.org/10.1145/3510003.3510057>.
- Cañete-Sifuentes, L., Monroy, R., & Medina-Pérez, M. A. (2021). A review and experimental comparison of multivariate decision trees. *IEEE Access*, 9, 110451-110479. DOI: 10.1109/ACCESS.2021.3102239.
- 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>.
- Costa, V. G., & Pedreira, C. E. (2023). Recent advances in decision trees: An updated survey. *Artificial Intelligence Review*, 56(5), 4765-4800. <https://doi.org/10.1007/s10462-022-10275-5>.
- Davis, E. (2015). Ethical guidelines for a superintelligence. *Artificial Intelligence*, 220, 121-124. <https://doi.org/10.1016/j.artint.2014.12.003>.
- Dhoni, P. (2023). Exploring the synergy between generative AI, data, and analytics in the modern age. Authorea Preprints. <https://doi.org/10.36227/techrxiv.24045792.v1>.
- Droutsas, I., Challinor, A. J., Deva, C. R., & Wang, E. (2022). Integration of machine learning into process-based modeling to improve the simulation of complex crop responses. in *silico Plants*, 4(2), diac017. <https://doi.org/10.1093/insilicoplants/diac017>.
- Elahi, M., Afolaranmi, S. O., Martinez Lastra, J. L., & Perez Garcia, J. A. (2023). A comprehensive literature review of the applications of AI techniques through the lifecycle of industrial equipment. *Discover Artificial Intelligence*, 3(1), 43. <https://doi.org/10.1007/s44163-023-00089-x>.
- Elmagarmid, A. K., Ipeiritos, P. G., & Verykios, V. S. (2006). Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering*, 19(1), 1-16. <https://doi.org/10.1109/TKDE.2007.250581>.
- Engen, M., Sandø, E., Sjølander, B. L. O., Arenberg, S., Gupta, R., & Goodwin, M. (2021). Farm-scale crop yield prediction from multi-temporal data using deep hybrid neural networks. *Agronomy*, 11(12), 2576. <https://doi.org/10.3390/agronomy11122576>.
- Everitt, T., Goertzel, B., & Potapov, A. (2017). *Artificial general intelligence. Lecture Notes in Artificial Intelligence*. Heidelberg: Springer. DOI: <https://doi.org/10.1007/978-3-319-63703-7>.
- Flasiński, M. (2016). *Introduction to artificial intelligence*. Springer. DOI: 10.1007/978-3-319-40022-8.



- Hajian-Tilaki K. (2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian journal of internal medicine*, 4(2), 627–635. PMID: 24009950.
- 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. DOI: https://doi.org/10.2991/978-94-91216-62-6_5.
- Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39, 261-283. <https://doi.org/10.1007/s10462-011-9272-4>
- 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>
- 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>.
- Omer, Z. M., & Shareef, H. (2022). Comparison of decision tree-based ensemble methods for prediction of photovoltaic maximum current. *Energy Conversion and Management: X*, 16, 100333. <https://doi.org/10.1016/j.ecmx.2022.100333>.
- 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>
- Plaia, A., Buscemi, S., Fürnkranz, J., & Mencía, E. L. (2022). Comparing boosting and bagging for decision trees of rankings. *Journal of Classification*, 1-22. <https://doi.org/10.1007/s00357-021-09397-2>.
- 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.
- 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>.
- Sarr, A. B., & Sultan, B. (2023). Predicting crop yields in Senegal using machine learning methods. *International Journal of Climatology*, 43(4), 1817-1838. <https://doi.org/10.1002/joc.7947>.
- 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. DOI: 10.23919/MIPRO.2019.8757088.
- Subasi, A. (2020). *Practical Machine Learning for Data Analysis*. Academic press. ISBN: 9780128213803.
- 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>.
- 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.
- Webb, T., Holyoak, K. J., & Lu, H. (2023). Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9), 1526-1541. <https://doi.org/10.1038/s41562-023-01659-w>.