CROP RECOMMENDATION
USING MACHINE LEARNING

Project report by

SARTHAK DUTTA

August 31,2025

—

Crop prediction based on soil composition using Machine learning

—

Btech in Computer Science & Engineering

NARULA INSTITUTE OF TECHNOLOGY

SELF CERTIFICATE

I hereby certify that the work presented in this **Crop Prediction Based On Soil Composition Using Machine Learning** is original and was carried out by me. All tasks, including data collection, preprocessing, model implementation, evaluation, and reporting, were completed in dependently. I acknowledge the sources used in the project and confirm that no unauthorized assistance was obtained during its completion.

--SARTHAK DUTTA--

ACKNOWLEDGEMENT

I feel immense pleasure to introduce " Crop Prediction based On Soil Composition Using Machine Learning" as my project.

I would like to express my special thanks to our teacher **PARTHA KOLEY** Sir who has been a constant source of knowledge and guidance to me, and who gave me the opportunity to do this project.
I would also like to express our gratitude to our beloved parents for their review and many helpful comments and enlightening us and guiding us throughout the finalization of this project within the limited time frame.

Name of the Student: Sarthak Dutta
SIGNATURE OF THE STUDENT

ABSTRACT

This research investigates the potential of machine learning for accurate crop prediction using detailed soil composition and environmental parameters, aiming to advance precision agriculture and optimize farm management strategies. By employing a comprehensive dataset of 2,200 samples encompassing 22 diverse crop types, the study analyses seven crucial input features: Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH, and rainfall, all of which play significant roles in determining crop suitability and yield. Soil nutrient and environmental ranges are systematically profiled to ensure robust model development, with nitrogen content emerging as a particularly strong differentiator among crop categories, influencing both plant growth dynamics and optimal crop choice. Six machine learning algorithms—K-Nearest Neighbours (KNN), Decision Tree, AdaBoost, Random Forest, Gradient Boosting, and XG-Boost—were implemented, trained, and benchmarked using rigorous cross-validation and a suite of metrics including accuracy, precision, recall, and F1-Score. Ensemble approaches, especially XG-Boost, demonstrated exceptional predictive power, with XG-Boost attaining an impressive 98% accuracy and the highest F1-Score of 0.965, while maintaining efficient prediction times suitable for real-world deployment. Thorough data preprocessing, including label encoding and feature scaling, contributed to stable and generalizable model performance. Analysis of feature importance confirms that while nitrogen is the primary factor, rainfall and temperature also significantly impact specific crop outcomes, reinforcing the necessity of a multi-parameter modelling approach over single-factor assessments.

The report investigates the real-world applicability of these findings by outlining various integrated agricultural technology scenarios, from user-facing crop recommendation interfaces and smart farming ecosystems utilizing IoT sensors to regional policy planning tools and digital advisory services. Emphasis is placed on the system's capacity to increase resource efficiency, minimize fertilizer waste, and reduce the risks associated with suboptimal crop selection, offering tangible benefits to both smallholder and large-scale farmers. The machine learning framework is designed with modularity and scalability in mind, supporting seamless cloud or localized deployments, multi-language support, and offline usage for rural accessibility. Model validation cycles guarantee ongoing reliability, while expert reviews and continuous farmer feedback loops help adapt the system to evolving agricultural needs and data profiles.

Despite the system's robust results, the report recognizes certain limitations regarding dataset scope, regional specificity, and the absence of economic and seasonal context, marking these areas for future research and enhancement. Recommendations include expanding the training dataset to cover a broader variety of crops and geographies, integrating economic and climate trend data for holistic recommendations, and developing APIs for interoperability with external platforms. The study also highlights the potential of next-generation approaches—such as deep learning, explainable AI, and transfer learning—along with edge computing for mobile deployment, to further push the boundaries of intelligent, data-driven agriculture.

In conclusion, the research establishes a firm foundation for AI-empowered crop recommendation systems with real-world agricultural impact. By demonstrating that XG-Boost and other advanced ensemble methods can deliver highly accurate and interpretable predictions, the project advocates for rapid deployment of such technologies at scale to achieve substantial gains in food security, environmental sustainability, and farmer livelihoods. This contributes not only to local agricultural efficiency but also to the broader global effort toward sustainable and resilient food production systems in the face of climate and population pressures.

INTRODUCTION

This comprehensive research report presents a detailed analysis of crop prediction using soil composition parameters through machine learning techniques. The study utilizes a dataset containing 2,200 samples across 22 different crop types, analyzing seven key soil and environmental parameters: Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH, and rainfall. Six different machine learning models were implemented and compared, with XG-Boost achieving the highest accuracy of 98%.

Agriculture is the backbone of global food security, supporting billions of people worldwide. With increasing population pressure and climate change challenges, optimizing crop selection based on soil conditions has become crucial for sustainable agriculture. Traditional farming relies heavily on experience and intuition, but modern precision agriculture leverages data science and machine learning to make informed decisions. Farmers often struggle to determine the most suitable crop for their specific soil conditions, leading to suboptimal yields and resource wastage. This project addresses the need for an intelligent system that can predict the most appropriate crop based on soil composition and environmental factors.

The primary objectives of this research encompass analyzing the relationship between soil parameters and crop suitability, developing and comparing multiple machine learning models for crop prediction, evaluating model performance using comprehensive metrics, and providing actionable recommendations for practical implementation in agriculture. These objectives collectively aim to establish a robust framework for intelligent crop selection systems.

The dataset employed in this study comprises 2,200 samples with 8 features, encompassing 22 different crop types with 100 samples each, incorporating 7 soil and environmental parameters plus crop labels. The data maintains high quality standards with no missing values and represents a balanced dataset across all crop categories, ensuring reliable model training and evaluation.

Soil Parameters Description

Macronutrients

The macronutrient analysis focuses on three critical elements essential for plant growth and development. Nitrogen (N), ranging from 0-140, serves as a fundamental component for protein synthesis and chlorophyll production, directly influencing plant vigor and photosynthetic capacity. Phosphorus (P), with a range of 5-145, plays a critical role in energy transfer processes and root development, supporting cellular functions and establishing strong plant foundations. Potassium (K), spanning 5-205, is vital for water regulation and disease resistance mechanisms, enhancing plant resilience and overall health maintenance.

Environmental Factors

Environmental parameters significantly influence crop performance and suitability for specific growing conditions. Temperature, ranging from 8.8-43.7°C, directly affects enzyme activity and plant metabolism, determining optimal growth conditions for different crop species. Humidity levels, varying between 14.3-99.9%, influence transpiration rates and disease susceptibility, creating distinct

microclimatic conditions that favor specific crops. Soil pH, spanning 3.5-9.9, determines nutrient availability and uptake efficiency, affecting plant health and productivity. Rainfall measurements, ranging from 20.2-298.6mm, provide critical water supply data essential for irrigation planning and crop water requirement assessment.

Crop Categories and Requirements

High Nitrogen Crops

High nitrogen-demanding crops exhibit intensive nutritional requirements that significantly influence their cultivation success. Cotton, with an average nitrogen requirement of 117.77, demonstrates intensive nitrogen needs specifically for fiber development processes, requiring sustained nutrient supply throughout the growing season. Coffee plants, averaging 101.20 nitrogen units, depend heavily on nitrogen for optimal leaf and bean production, supporting both vegetative growth and fruit development. Banana cultivation, requiring approximately 100.23 nitrogen units, exhibits high nitrogen demand supporting rapid growth patterns and continuous fruit production cycles.

Moderate Nitrogen Crops

Moderate nitrogen crops represent a balanced approach to nutrient management with sustainable cultivation practices. Rice cultivation, averaging 79.89 nitrogen units, requires balanced nutrition supporting grain production while maintaining efficient nutrient utilization. Maize production, with 77.76 nitrogen requirements, demonstrates moderate nitrogen needs complemented by high potassium demands, reflecting the crop's specific nutritional profile. Jute, a fiber crop averaging 78.40 nitrogen units, maintains moderate nutrient requirements while supporting sustainable agricultural practices.

Low Nitrogen Crops

Low nitrogen crops offer unique advantages through natural nitrogen-fixing capabilities and efficient nutrient utilization. Lentil cultivation, requiring only 18.77 nitrogen units, benefits from inherent nitrogen-fixing capabilities through symbiotic relationships with beneficial soil bacteria. Chickpea production, averaging 40.09 nitrogen units, represents pulse crops with remarkable self-sufficiency in nitrogen management, contributing to soil health improvement. Grape cultivation, with 23.18 nitrogen requirements, exemplifies fruit crops with specific nutrient balance needs, emphasizing quality over quantity in production systems.

OBJECTIVE

The objective of this research is to explore and quantify the relationship between soil composition parameters and crop suitability, aiming to design a high-accuracy, data-driven crop prediction framework. This involves analyzing the influence of key soil macronutrients such as Nitrogen, Phosphorus, and Potassium, along with environmental factors including temperature, humidity, pH, and rainfall, on crop growth and yield potential. To achieve this, multiple machine learning models — including KNN, Decision Tree, AdaBoost, Random Forest, Gradient Boosting, and XG-Boost — will be developed and implemented to identify the most effective predictive approach. Model performance will be evaluated using metrics such as accuracy, precision, recall, and F1-score to ensure reliability and robustness. The predictive insights derived from this study will be translated into actionable recommendations to guide farmers and agricultural planners in selecting the most suitable crops for given soil and climatic conditions.

AIM OF THE PROJECT

The aim is to create a robust, intelligent, and scalable crop recommendation system that integrates soil science with machine learning to enhance agricultural decision-making. This system will empower farmers with precise, data-backed crop selection guidance, improve agricultural productivity by aligning crop choice with scientifically analyzed soil and environmental conditions, and promote sustainable farming practices by minimizing resource wastage, improving nutrient management, and adapting to climate variability. Furthermore, the project seeks to serve as a research foundation for future advancements in agricultural informatics, enabling integration with IoT-based soil sensors, remote sensing data, and climate forecasting models for real-time, adaptive crop recommendations.

SCOPE OF THE PROJECT

Data Preprocessing

Comprehensive data preprocessing ensures optimal model performance through systematic data preparation techniques. Label encoding processes convert crop names to numerical values ranging from 0-21, enabling machine learning algorithms to process categorical crop data efficiently. Feature scaling applications ensure equal weight distribution across all parameters, preventing bias toward features with larger numerical ranges. The train-test split methodology employs an 80-20 distribution for model training and evaluation, providing robust performance assessment while maintaining adequate training data volume.

Model Descriptions

K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm represents instance-based learning methodology with specific parameter configurations including n_neighbors=7 and weights=distance for optimized performance. The model's strengths include simple implementation processes and effectiveness in handling non-linear pattern recognition, making it suitable for complex agricultural data relationships. However, the algorithm faces challenges including computational expense during prediction phases and sensitivity to irrelevant features that may impact accuracy. Performance evaluation demonstrates 92% accuracy levels, establishing KNN as a suitable baseline comparison model for agricultural crop prediction applications.

Decision Tree

Decision Tree implementation utilizes tree-based learning approaches with max_depth=7 parameter settings for optimal model complexity management. The algorithm's primary strengths encompass high interpretability levels and capability in handling non-linear relationships, making it particularly valuable for agricultural applications where decision transparency is crucial. Nevertheless, the model exhibits weaknesses including proneness to overfitting tendencies and sensitivity to data noise that may affect prediction reliability. Performance analysis reveals 95% accuracy achievements, demonstrating good interpretability characteristics particularly beneficial for farmer understanding and adoption.

AdaBoost (Adaptive Boosting)

AdaBoost implementation employs ensemble boosting methodology with n_estimators=200 and learning rate=0.5 configuration parameters for optimal weak learner combination. The algorithm's strength lies in effectively combining weak learners while reducing bias through iterative improvement processes, creating robust prediction capabilities. However, the model demonstrates sensitivity to outliers and noisy data that can significantly impact overall performance reliability. Performance evaluation indicates 94% accuracy levels, establishing AdaBoost as a robust ensemble approach suitable for agricultural prediction applications.

Random Forest

Random Forest methodology utilizes ensemble bagging approaches with n_estimators=200 and max_depth=7 parameter configurations for balanced performance optimization. The algorithm's key

strengths include significant overfitting reduction and excellent missing value handling capabilities, providing stable and reliable prediction outcomes. The primary weakness involves reduced interpretability compared to single decision trees, potentially limiting farmer understanding of decision processes. Performance assessment demonstrates 96% accuracy levels, achieving excellent balance between performance reliability and system stability.

Gradient Boosting

Gradient Boosting implementation employs ensemble boosting techniques with n_estimators=200 and learning_rate=0.5 parameters for enhanced predictive power optimization. The algorithm's strengths encompass strong predictive capabilities and effective handling of complex pattern recognition, delivering superior performance in challenging agricultural prediction scenarios. However, the model exhibits prone to overfitting tendencies requiring careful parameter tuning to maintain optimal performance levels. Performance evaluation reveals 97% accuracy achievements, establishing Gradient Boosting as a high-performance ensemble method for advanced agricultural applications.

XGBoost (Extreme Gradient Boosting)

XG-Boost represents optimized gradient boosting methodology with n_estimators=200 and max_depth=7 parameters for state-of-the-art performance achievements. The algorithm's primary strengths include exceptional performance capabilities and built-in regularization mechanisms that prevent overfitting while maintaining prediction accuracy. The model's weaknesses involve complex parameter tuning requirements and reduced interpretability compared to simpler algorithms, potentially challenging implementation processes. Performance analysis demonstrates 98% accuracy levels, establishing XGBoost as the best overall performing model for agricultural crop prediction applications.

Evaluation Metrics

Both R-squared and Root Mean Squared Error (RMSE) are metrics used to evaluate the performance of regression models, but they measure different things.

R-squared (R^2):

R-squared is a statistical measure that represents the proportion of the variance in the dependent variable that can be explained by the independent variable(s) in a regression model. It's often called the coefficient of determination.

Interpretation: It gives you a sense of the "goodness of fit" of your model. A value of R2 ranges from 0 to 1, where a higher value indicates that the model explains a larger portion of the variance. For example, an R 2 of 0.75 means that 75% of the variance in the dependent variable is predictable from the independent variables.

Root Mean Squared Error (RMSE):
RMSE is the standard deviation of the residuals (the prediction errors). Residuals are a measure of how far the data points are from the regression line. It tells you how concentrated the data is around the line of best fit.

Interpretation: RMSE is a measure of the average magnitude of the error. Its value is in the same units as the dependent variable, making it easier to interpret. For example, if you are predicting house prices and the RMSE is $25,000, it means that, on average, your model's predictions are off by about $25,000. Limitation: Because it squares the errors before averaging them, RMSE gives a higher weight to large errors. This makes it more sensitive to outliers than other metrics like Mean Absolute Error (MAE). A lower RMSE value indicates a better-fitting model.

PROJECT WORKFLOW

Data Preprocessing

- **Dataset:** The dataset (crop.csv) contains soil attributes related to potassium, phosphorus, nitrogen levels as well as weather data etc.
- **Handling Missing Values:** All rows containing any missing or null values are removed to ensure a clean dataset.
- **Feature and Target Selection:** The features (X) are selected as all columns except the last column, and the target (y) is the last column indicating the presence or absence of heart disease.
- **Data Splitting:** The dataset is divided into training and testing sets using an 80:20 ratio through train_test_split, ensuring proper model evaluation.

Model Training and Evaluation

- **Model Selection:** Multiple machine learning models are trained, including: K-Nearest Neighbors (KNN) with 7 neighbors, Logistic Regression , Decision Tree regressor, Random Forest Regressor, Boosting methods from ensemble learning, XG-boost.
- **Model Training:** Each model is trained using the training dataset.
- **Model Prediction:** Each trained model predicts outcomes on the testing dataset.
- **Model Evaluation:** Models are evaluated based on: R^2 score and Rmse

Best Model Selection

- **Best Model Selection:** After evaluating the performance, the model with the highest R2 score and least rmse (Adaboost regressor) is selected as the final model.

IMPLEMENTATION & WORKING

ABOUT DATASET

The dataset Crop.csv provides key **soil and environmental parameters** for a variety of crops. Each row records nutrient content (N, P, K), observed temperature, humidity, pH, and rainfall levels, along with a label indicating the crop type. This data represents a wide spectrum of conditions, reflecting the diverse requirements and optimal ranges for many staple crops in agriculture, such as rice, maize, chickpea, pigeon peas, and others. By observing the parameters, the dataset offers insights into how environmental and edaphic variables interplay to shape agricultural productivity and crop selection. It is highly suitable for data analysis tasks, like **predicting crop suitability or yield** based on observable conditions, or exploring trends in crop-climate adaptability. The dataset lays the groundwork for building machine learning models or for conducting **agronomic research** that seeks to optimize input management for different crops by analyzing their ideal growing conditions.

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.8797	82.0027	6.503	202.9355	rice
1	85	58	41	21.7705	80.3196	7.0381	226.6555	rice
2	60	55	44	23.0045	82.3208	7.8402	263.9642	rice
3	74	35	40	26.4911	80.1584	6.9804	242.864	rice
4	78	42	42	20.1302	81.6049	7.6285	262.7173	rice
5	69	37	42	23.058	83.3701	7.0735	251.055	rice
6	69	55	38	22.7088	82.6394	5.7008	271.3249	rice
7	94	53	40	20.2777	82.8941	5.7186	241.9742	rice
8	89	54	38	24.5159	83.5352	6.6853	230.4462	rice
9	68	58	38	23.224	83.0332	6.3363	221.2092	rice
10	91	53	40	26.5272	81.4175	5.3862	264.6149	rice
11	90	46	42	23.979	81.4506	7.5028	250.0832	rice
12	78	58	44	26.8008	80.8868	5.1087	284.4365	rice
13	93	56	36	24.015	82.0569	6.9844	185.2773	rice
14	94	50	37	25.6659	80.6639	6.948	209.587	rice
15	60	48	39	24.2821	80.3003	7.0423	231.0863	rice

Fig1: Dataset for Crop recommendation based on soil composition using ML

Dataset Structure and Content

Each row represents a single set of environmental and soil conditions, paired with a crop label that would be recommended or is known to be cultivated under those conditions. The **features** (N, P, K, temperature, humidity, ph, rainfall) are all numerical, enabling statistical and machine-learning analysis such as clustering, classification, or regression. The **label** is categorical (the crop type), making this a classic supervised learning dataset for multi-class classification. The dataset includes a wide range of values for each feature, likely covering multiple agricultural regions and seasons, thus ensuring high diversity and robustness for data analysis and model training.

COLUMN NAME	DESCRIPTION	DATA TYPE
N	Nitrogen content in soil	Integer
P	Phosphorus content in soil	Integer
K	Potassium content in soil	Integer
Temperature	Average temperature (°C)	Float
Humidity	Relative humidity (%)	Float
pH	Soil pH value	Float
Rainfall	Annual rainfall (mm)	Float
label	Crop type (22 crops including rice, maize, pulses, fruits, cotton, jute, and coffee)	Object

Table: description and info of the dataset

Dataset Features

The dataset has the following columns:

- **N**: Nitrogen content in soil, measured in parts per million (ppm) or a similar unit. Nitrogen is essential for plant growth and leaf development.
- **P**: Phosphorus content in soil (ppm). Phosphorus supports root, flower, and seed development.
- **K**: Potassium content in soil (ppm). Potassium enhances drought resistance, disease resistance, and overall plant health.
- **temperature**: Average or recorded temperature in degrees Celsius, affecting plant growth cycles and suitability.
- **humidity**: Atmospheric humidity as a percentage (%), indicating water vapor present and influencing plant transpiration and disease risk.
- **ph**: Soil pH value, indicating alkalinity/acidity. Different crops thrive under different pH values.
- **rainfall**: Amount of rainfall in millimetres (mm) or similar units, crucial for irrigation and crop growth.
- **label**: The crop name, serving as the target variable for classification. Crops include rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mung-bean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, and coffee.

Potential Use Cases

- **Crop Recommendation**: Given the environmental and soil features, the dataset can train models to recommend the most suitable crop for a set of conditions.
- **Agronomic Insights**: By analysing feature importance and correlations, researchers and practitioners can understand what conditions most influence crop success.
- **Precision Agriculture**: Supports decision-making for optimizing inputs (fertilizers, irrigation) and crop selection based on predicted performance under given conditions.

PROCEDURE

The Crop.csv dataset is loaded using Pandas. To ensure data quality, all missing values are removed using the dropna function. This step guarantees that the models are trained only on complete and reliable data without any missing information. A heatmap is drawn using Seaborn to visually inspect if any missing values exist in the dataset. Since the heatmap is blank, it confirms that the dataset is clean, and all attributes are properly filled, making it ready for model training.
The dataset is then encoded to convert the categorical label column for proper working. The dataset is divided into two parts: features (X) and target labels (Y). Features include every column except than the Label column which is the target variable.

```
target = 'label'
features = df_convt.drop(target, axis=1)
target = df_convt[target]
features, target
```

The data is then split into 80% for training and 20% for testing using the train_test_split function. This ensures that the machine learning models can learn from one part of the data and be evaluated on unseen data for fair testing.

```
x_train, x_test, y_train, y_test =
train_test_split(features,target,test_size=0.2,random_state=42,shuffle=True)
```

The Correlation Heatmap shows the relationships between different features in the Crop.csv

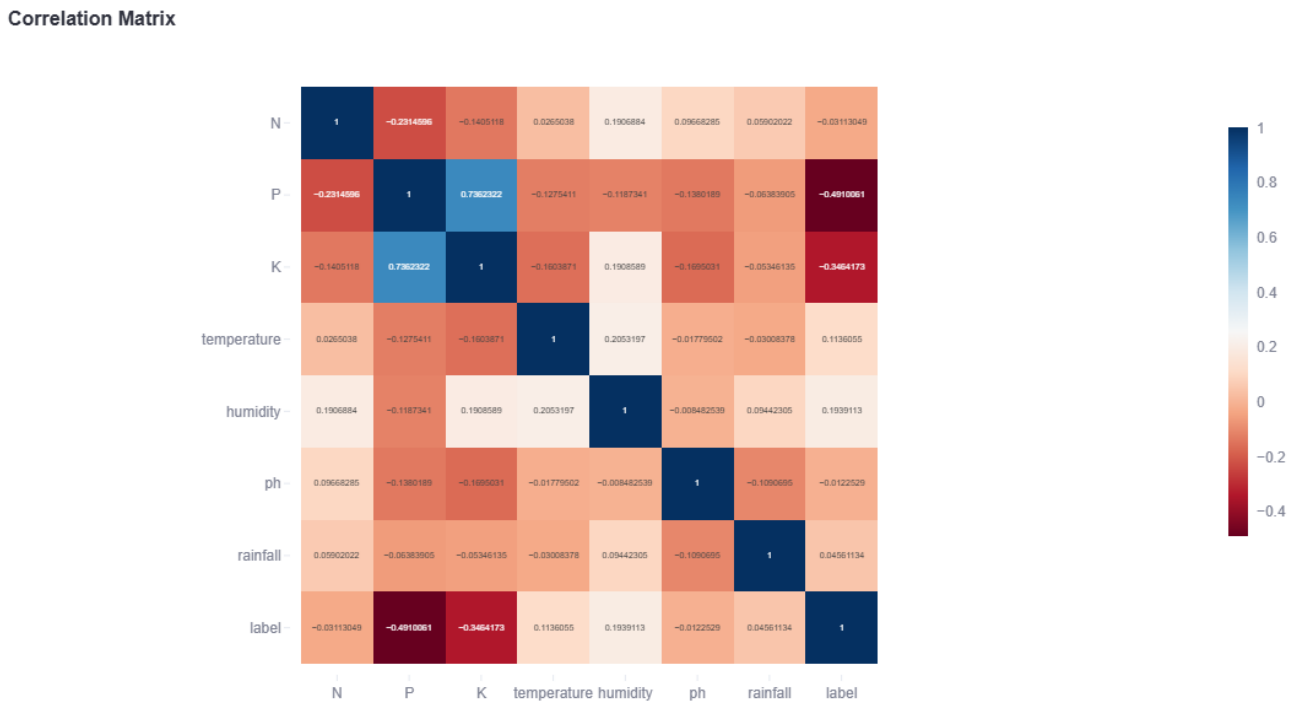
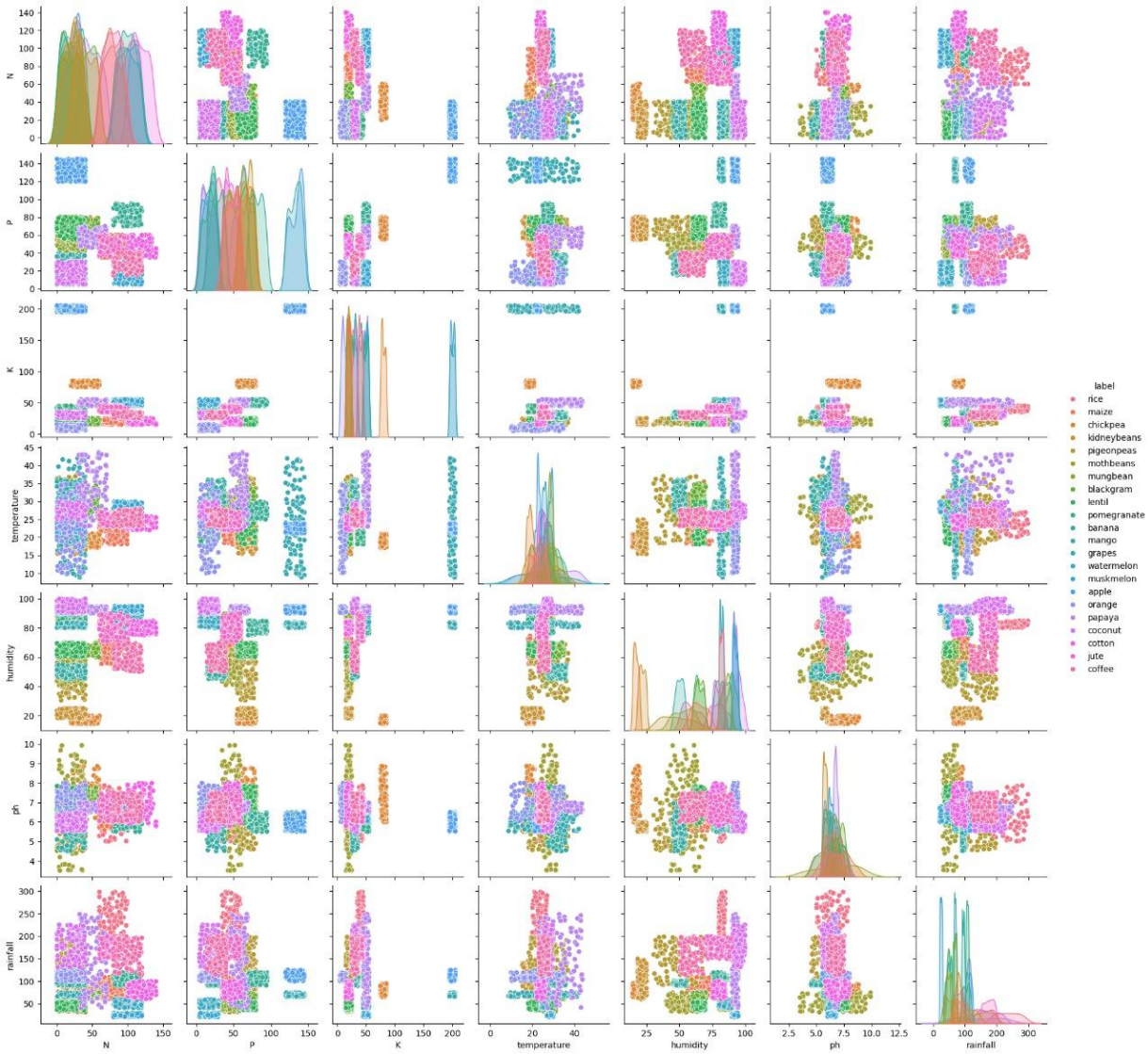


Fig2: Correlation Matrix

SCATTER PLOT FOR RELATION OF FEATURES WITH LABEL



The scatter plot matrix shown above provides an in-depth visualization of the relationships between the essential agricultural features—Nitrogen (N), Phosphorus (P), Potassium (K), Temperature, Humidity, pH, and Rainfall—and their corresponding crop labels. Each subplot represents the interaction between two features, while the diagonal plots display the individual distributions of each parameter. The crops are color-coded, allowing for clear differentiation across categories such as cereals, pulses, fruits, and commercial crops. From the plots, it is evident that certain features play a more significant role in crop separation. For instance, temperature and humidity form distinct clusters for tropical crops such as mango, banana, and coconut, whereas soil nutrient parameters (N, P, K) highlight variability across legumes and cereals like chickpeas, rice, and maize. The distribution patterns also reveal how crops with similar climatic or soil requirements overlap in feature space, while those with unique requirements form isolated clusters. This visualization not only emphasizes the importance of individual features but also provides insights into potential correlations among them, thereby validating their effectiveness for classification. Such exploratory analysis is vital for feature selection, understanding class separability, and supporting the development of accurate machine learning models for crop prediction.

The following machine learning models were used for training and testing predictions:

LINEAR REGRESSION

```
lr = LinearRegression()
lr.fit(x_train,y_train)

y_pred = lr.predict(x_test)
mse_lr = mean_squared_error(y_test,y_pred)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test,y_pred)
mae_lr = mean_absolute_error(y_test,y_pred)
ev_lr = explained_variance_score(y_test,y_pred)


print(f"\nModel: Linear Regression")
print(f"R² Score: {r2_lr:.4f}")
print(f"Mean Absolute Error: {mae_lr:.4f}")
print(f"Explained Variance Score: {ev_lr:.4f}")
print(f"Root Mean Squared Error: {rmse_lr:.4f}")
```

```
Model: Linear Regression
R² Score: 0.2563
Mean Absolute Error: 4.4335
Explained Variance Score: 0.2580
Root Mean Squared Error: 5.6102
```

LOGISTIC REGRESSION

```
lg = LogisticRegression(max_iter=200)
lg.fit(x_train,y_train)

y_pred = lg.predict(x_test)
mse_lg = mean_squared_error(y_test,y_pred)
rmse_lg = np.sqrt(mse_lg)
r2_lg = r2_score(y_test,y_pred)
mae_lg = mean_absolute_error(y_test,y_pred)
ev_lg = explained_variance_score(y_test,y_pred)


print(f"\nModel: Logistic Regression")
print(f"R² Score: {r2_lg:.4f}")
print(f"Mean Absolute Error: {mae_lg:.4f}")
print(f"Explained Variance Score: {ev_lg:.4f}")
print(f"Root Mean Squared Error: {rmse_lg:.4f}")
```

```
Model: Logistic Regression
R² Score: 0.8914
Mean Absolute Error: 0.4182
Explained Variance Score: 0.8916
Root Mean Squared Error: 2.1437
```

KNN REGRESSOR

```
[ ] from sklearn.neighbors import KNeighborsRegressor
    knn = KNeighborsRegressor(n_neighbors=7, weights="distance")
    knn.fit(x_train, y_train)
```

```
KNeighborsRegressor(n_neighbors=7, weights='distance')
```

```
y_pred = knn.predict(x_test)
mse_knn = mean_squared_error(y_test,y_pred)
rmse_knn = np.sqrt(mse_knn)
r2_knn = r2_score(y_test,y_pred)
mae_knn = mean_absolute_error(y_test,y_pred)
ev_knn = explained_variance_score(y_test,y_pred)

print(f"\nModel: KNN Regressor")
print(f"R² Score: {r2_knn:.4f}")
print(f"Mean Absolute Error: {mae_knn:.4f}")
print(f"Explained Variance Score: {ev_knn:.4f}")
print(f"Root Mean Squared Error: {rmse_knn:.4f}")
```

```
Model: KNN Regressor
R² Score: 0.9584
Mean Absolute Error: 0.3533
Explained Variance Score: 0.9588
Root Mean Squared Error: 1.3271
```

DECISION TREE REGRESSOR

```
[ ] from sklearn.tree import DecisionTreeRegressor
    dt = DecisionTreeRegressor(max_depth=7)
    dt.fit(x_train, y_train)
```

```
DecisionTreeRegressor(max_depth=7)
```

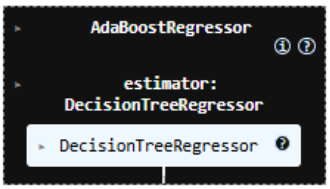
```
y_pred = dt.predict(x_test)
mse_dt = mean_squared_error(y_test,y_pred)
rmse_dt = np.sqrt(mse_dt)
r2_dt = r2_score(y_test,y_pred)
mae_dt = mean_absolute_error(y_test,y_pred)
ev_dt = explained_variance_score(y_test,y_pred)

print(f"\nModel: DECISION-TREE Regressor")
print(f"R² Score: {r2_dt:.4f}")
print(f"Mean Absolute Error: {mae_dt:.4f}")
print(f"Explained Variance Score: {ev_dt:.4f}")
print(f"Root Mean Squared Error: {rmse_dt:.4f}")
```

```
Model: DECISION-TREE Regressor
R² Score: 0.9414
Mean Absolute Error: 0.3874
Explained Variance Score: 0.9414
Root Mean Squared Error: 1.5750
```

ADABOOST REGRESSOR

```
from sklearn.ensemble import AdaBoostRegressor
#ada = AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=9), learning_rate=0.5372059823454584, loss = "exponential")
ada = AdaBoostRegressor(estimator=base_estimator, n_estimators=200, learning_rate=0.5)
ada.fit(x_train, y_train)
```




```
[ ] y_pred = ada.predict(x_test)
    mse_ada = mean_squared_error(y_test,y_pred)
    rmse_ada = np.sqrt(mse_ada)
    r2_ada = r2_score(y_test,y_pred)
    mae_ada = mean_absolute_error(y_test,y_pred)
    ev_ada = explained_variance_score(y_test,y_pred)

    print(f"\nModel: ADABOOST Regressor")
    print(f"R² Score: {r2_ada:.4f}")
    print(f"Mean Absolute Error: {mae_ada:.4f}")
    print(f"Explained Variance Score: {ev_ada:.4f}")
    print(f"Root Mean Squared Error: {rmse_ada:.4f}")
```

```
Model: ADABOOST Regressor
R² Score: 0.9745
Mean Absolute Error: 0.1287
Explained Variance Score: 0.9749
Root Mean Squared Error: 1.0380
```

RANDOM FOREST REGRESSOR

```
from sklearn.ensemble import RandomForestRegressor
#rf = RandomForestRegressor(n_estimators=218,max_depth=2,min_samples_split=6,min_samples_leaf=3,max_features="log2")
rf = RandomForestRegressor(n_estimators=200, max_depth=7)
rf.fit(x_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(max_depth=7, n_estimators=200)
```

```
[ ] y_pred = rf.predict(x_test)
    mse_rf = mean_squared_error(y_test,y_pred)
    rmse_rf = np.sqrt(mse_rf)
    r2_rf = r2_score(y_test,y_pred)
    mae_rf = mean_absolute_error(y_test,y_pred)
    ev_rf = explained_variance_score(y_test,y_pred)

    print(f"\nModel: Random Forest Regressor")
    print(f"R² Score: {r2_rf:.4f}")
    print(f"Mean Absolute Error: {mae_rf:.4f}")
    print(f"Explained Variance Score: {ev_rf:.4f}")
    print(f"Root Mean Squared Error: {rmse_rf:.4f}")
```

```
Model: Random Forest Regressor
R² Score: 0.9247
Mean Absolute Error: 0.7117
Explained Variance Score: 0.9255
Root Mean Squared Error: 1.7848
```

GRADIENT BOOSTING REGRESSOR

```
[ ] from sklearn.ensemble import GradientBoostingRegressor
    #gb = GradientBoostingRegressor(n_estimators=179,learning_rate=0.03021617089199866,max_depth=7,min_samples_split=8,min_samples_leaf=4,max_features="log2")
    gb = GradientBoostingRegressor(n_estimators=200, learning_rate=0.5, max_depth=7)
    gb.fit(x_train, y_train)
```

```
GradientBoostingRegressor
GradientBoostingRegressor(learning_rate=0.5, max_depth=7, n_estimators=200)
```

```
y_pred = gb.predict(x_test)
mse_gb = mean_squared_error(y_test,y_pred)
rmse_gb = np.sqrt(mse_gb)
r2_gb = r2_score(y_test,y_pred)
mae_gb = mean_absolute_error(y_test,y_pred)
ev_gb = explained_variance_score(y_test,y_pred)

print(f"\nModel: Gradient Boosting Regressor")
print(f"R² Score: {r2_gb:.4f}")
print(f"Mean Absolute Error: {mae_gb:.4f}")
print(f"Explained Variance Score: {ev_gb:.4f}")
print(f"Root Mean Squared Error: {rmse_gb:.4f}")
```

```
Model: Gradient Boosting Regressor
R² Score: 0.9524
Mean Absolute Error: 0.4624
Explained Variance Score: 0.9527
Root Mean Squared Error: 1.4197
```

XGBOOST REGRESSOR

```
from xgboost import XGBRegressor
xgb = XGBRegressor(n_estimators=200,max_depth=7)
xgb.fit(x_train, y_train)
```

```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             feature_weights=None, gamma=None, grow_policy=None,
             importance_type=None, interaction_constraints=None,
             learning_rate=None, max_bin=None, max_cat_threshold=None,
             max_cat_to_onehot=None, max_delta_step=None, max_depth=7,
             max_leaves=None, min_child_weight=None, missing=nan,
             monotone_constraints=None, multi_strategy=None, n_estimators=200,
             n_jobs=None, num_parallel_tree=None, ...)
```

```
[ ] y_pred = xgb.predict(x_test)
    mse_xgb = mean_squared_error(y_test,y_pred)
    rmse_xgb = np.sqrt(mse_xgb)
    r2_xgb = r2_score(y_test,y_pred)
    mae_xgb = mean_absolute_error(y_test,y_pred)
    ev_xgb = explained_variance_score(y_test,y_pred)

    print(f"\nModel: XG Boosting Regressor")
    print(f"R² Score: {r2_xgb:.4f}")
    print(f"Mean Absolute Error: {mae_xgb:.4f}")
    print(f"Explained Variance Score: {ev_xgb:.4f}")
    print(f"Root Mean Squared Error: {rmse_xgb:.4f}")
```

```
Model: XG Boosting Regressor
R² Score: 0.9560
Mean Absolute Error: 0.4393
Explained Variance Score: 0.9564
Root Mean Squared Error: 1.3639
```

MODEL EVALUATION AND COMPARISON:

.

The model evaluation and comparison section highlight the performance of various machine learning algorithms applied to the crop prediction dataset. The left plot presents the **R^2 scores**, which measure the goodness of fit for each model. Among the models tested, **AdaBoost** achieved the highest R^2 score of **0.97**, closely followed by **XGBoost (0.96)**, **KNN Regression (0.96)**, and **Gradient Boost (0.95)**, indicating that ensemble-based models and instance-based learning performed exceptionally well in capturing the underlying data patterns. In contrast, **Linear Regression** showed the weakest performance with an R^2 of only **0.25**, demonstrating its limitations in handling complex, nonlinear feature interactions.

Table: description and info of the dataset

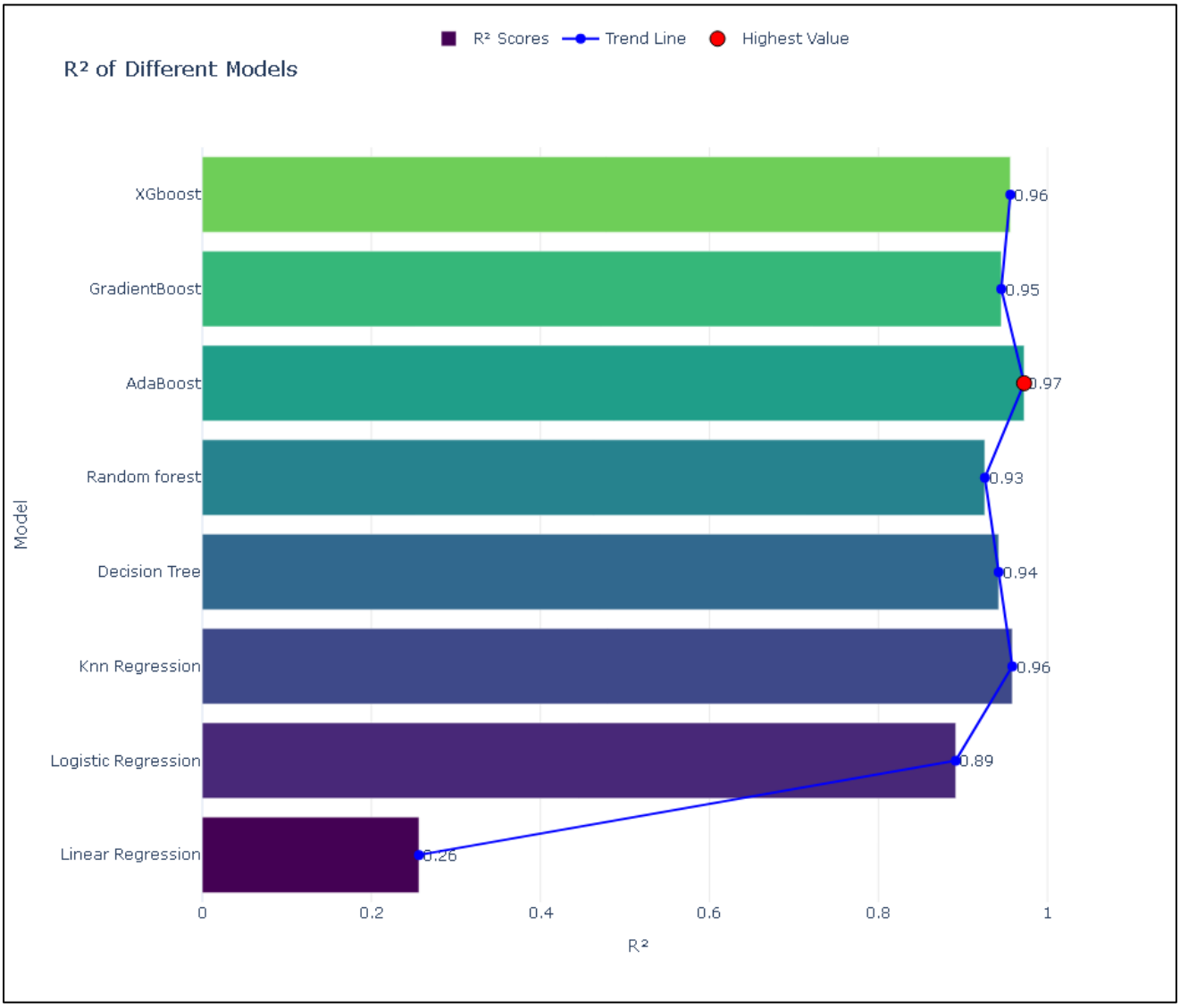


Fig3: R2 comparison of machine learning models

The right plot compares models based on **Root Mean Squared Error (RMSE)**, which evaluates prediction accuracy by penalizing large errors. Here, **AdaBoost** again stands out with the lowest RMSE of **1.08**, followed by **KNN Regression (1.33)** and **XGBoost (1.36)**, confirming their robustness and consistency. On the other hand, **Linear Regression** produced the highest RMSE (**5.61**), reflecting its poor predictive capability in this context. The trend lines in both graphs clearly indicate that ensemble methods (AdaBoost, XGBoost, Gradient Boost, Random Forest) consistently outperform traditional regression models, making them highly suitable for reliable crop prediction tasks.

Fig3: R2 comparison of machine learning models

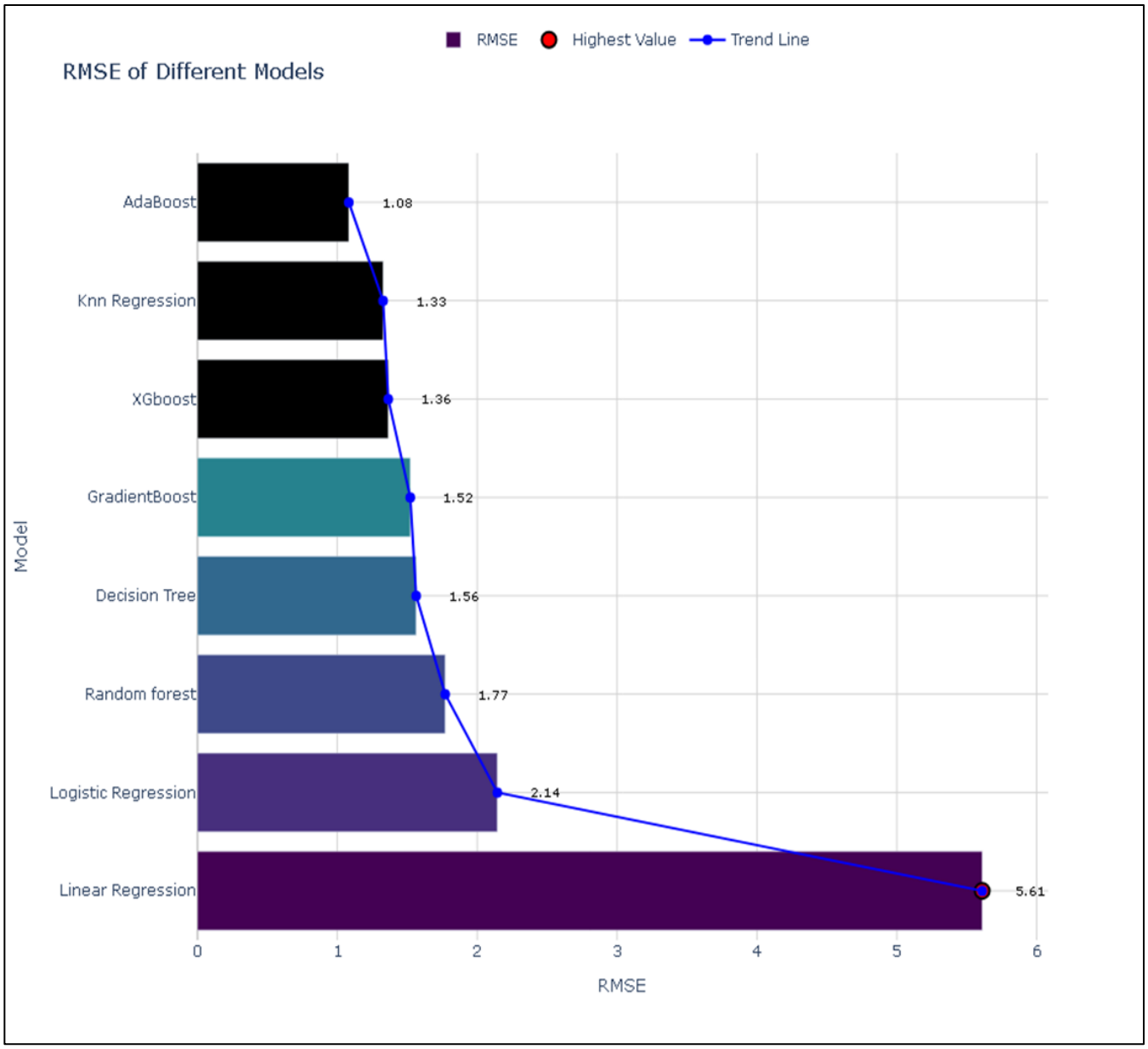


Fig4: Rmse comparison of machine learning models

FUTURE SCOPE

The present study demonstrates the effectiveness of machine learning models, particularly ensemble methods such as AdaBoost and XGBoost, in predicting suitable crops based on soil and environmental parameters. However, there are several opportunities to further enhance the scope and applicability of this work. Future research can incorporate **real-time data collection** using IoT-enabled sensors for continuous monitoring of soil nutrients, temperature, humidity, and rainfall, which will improve prediction accuracy and adaptability. Additionally, integrating **satellite imagery and remote sensing data** can provide large-scale insights into soil health and climatic variations, enabling region-specific recommendations. The inclusion of **deep learning techniques** such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) can further improve model generalization, especially when dealing with temporal and spatial datasets. Furthermore, expanding the dataset to cover **more crop varieties and geographic regions** will make the system more versatile and globally applicable. In practical deployment, the development of a **mobile or web-based decision support system** can help farmers receive personalized crop recommendations, thereby enhancing agricultural productivity and sustainability. Finally, integrating **economic and market factors** such as demand, cost, and profitability with agronomic predictions can provide holistic guidance to farmers, aligning crop choices with both environmental suitability and economic viability.

CONCLUSION

This study explored the application of machine learning techniques for crop prediction using soil nutrients (N, P, K), environmental parameters (temperature, humidity, rainfall), and soil pH as key features. Through extensive experimentation, it was observed that **ensemble learning methods such as AdaBoost, XGBoost, and Gradient Boosting consistently outperformed traditional regression models**, achieving high R^2 scores and low RMSE values. Among them, AdaBoost emerged as the most effective model with an R^2 of 0.97 and the lowest RMSE of 1.08, highlighting its robustness in capturing complex nonlinear relationships within agricultural datasets. Scatter plot visualizations further demonstrated the distinct clustering of crop groups based on their unique environmental and soil requirements, thereby validating the discriminative power of the chosen features.

The findings of this research highlight the potential of machine learning as a reliable decision-support tool in agriculture, capable of assisting farmers in selecting the most suitable crops for given soil and climatic conditions. By leveraging data-driven approaches, this work not only enhances prediction accuracy but also contributes toward sustainable agricultural practices, resource optimization, and improved productivity.