Department of Computer Science

Indian Institute of Technology Jodhpur

Program: Postgraduate Diploma in Data Engineering

Trimester – II

Subject: Machine Learning (CSL7620)

Project: CropPredictorX: ML for Optimal Crop Selection

Student Name	Roll Number	Email ID	GitHub Links
Harshita Gupta	G24AI2017	G24AI2017@iitj.ac.in	https://github.com/gpt-sarthak/ML_Crop_recommendation
Kaushal Kushwaha	G24AI2098	G24AI2098@iitj.ac.in	
Saransh Punia	G24AI2093	G24AI2093@iitj.ac.in	
Sarthak Gupta	G24AI2057	G24AI2057@iitj.ac.in	
Ankit Kumar Bhatnagar	G24AI2022	G24AI2022@iitj.ac.in	

Table of Contents

Table of Contents 2

Abstract 3

1. Introduction 4

1.1 Background and Context 4

1.2 Problem Statement 4

1.3 Objectives 4

2. Literature Review 5

3. Data Description 6

4. Exploratory Data Analysis (EDA) 7

4.1 Descriptive Statistics 7

4.3 Distribution Plots 7

4.2 Outlier Detection 8

5. Methodology 8

5.1 Data Preprocessing 8

5.2 Classifiers 9

5.2.1 Logistic Regression 9

5.2.2 Naive Bayes (GaussianNB) 10

5.2.3 Support Vector Machine (SVM) 11

5.2.4 K-Nearest Neighbors (KNN) 12

5.2.5 Decision Tree 14

5.2.6 Random Forest 15

5.2.7 Bagging Classifier 16

5.2.8 AdaBoost 18

5.2.9 Gradient Boosting 19

5.2.10 Extra Trees 21

6. Implementation Details 22

6.1 Development Environment 22

6.2 Project Workflow 23

6.3 GitHub Link for Code ____ 23

6.4 Reproducibility ____ 23

7. Results ____ 23

8. Discussion ____ 25

8.1 Interpretation of Results ____ 26

8.2 Comparison of Classifiers ____ 26

8.3 Strengths of the Approach ____ 26

8.4 Limitations ____ 27

8.5 Ethical Considerations ____ 27

8.6 Broader Implications ____ 27

9. Conclusion ____ 28

10. References ____ 28

Abstract

Precision agriculture integrates data science, agronomy, and decision-support systems to optimize resource utilization and enhance crop productivity. This project, **CropPredictorX: Machine Learning for Optimal Crop Selection**, presents the design and evaluation of a supervised learning framework that recommends suitable crops based on soil and environmental parameters.

A dataset of 2,200 observations encompassing macro-nutrient concentrations (Nitrogen, Phosphorus, Potassium), soil pH, temperature, humidity, and rainfall was analyzed. Exploratory data analysis assessed variable distributions, interdependencies, and outliers. Four classifiers—Random Forests, Support Vector Machines, K-Nearest Neighbors, and Decision Trees—were developed with hyperparameter tuning via grid search and stratified k-fold cross-validation. The Random Forest classifier achieved the highest predictive performance, attaining ~99% accuracy across 22 crop categories. Feature importance analysis identified Nitrogen, Potassium, and pH as the most influential predictors.

The modeling pipeline, including preprocessing, feature scaling, and evaluation, is documented with figures and confusion matrices illustrating results. While the findings demonstrate the feasibility of machine learning–based crop recommendations, limitations include the lack of temporal and geospatial data. Future work will explore integrating satellite imagery, seasonal dynamics, and end-user tools for personalized recommendations.

Overall, *CropPredictorX* contributes a reproducible framework that highlights the potential of supervised learning to advance precision agriculture, support sustainable planning, and enhance food security.

1. Introduction

1.1 Background and Context

Agriculture sustains the livelihoods of billions and is foundational to economic and social stability. However, traditional crop selection practices often rely on local heuristics, experience, and generalized recommendations, potentially leading to suboptimal yields and resource inefficiency. Recent advances in machine learning provide opportunities to transform agricultural practices by leveraging data-driven insights.

1.2 Problem Statement

Identifying the most suitable crop for cultivation in a given region is a complex, multivariate problem influenced by soil characteristics, climatic factors, and seasonal variability. The challenge is to construct a predictive model that can accurately recommend crops based on quantifiable features.

1.3 Objectives

This project aims to:

- ♦ To build a robust model to give correct and accurate prediction of crop sustainability in each state for the soil type and climatic conditions.
- ♦ Provide recommendation of the best suitable crops in the area so that the farmer does not incur any losses

- ♦ Evaluate and compare model performance across classifiers.
- ♦ Document a reproducible pipeline for future research and practical applications.

2. Literature Review

Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique Authors: Rakesh Kumar, M.P. Singh, Prabhat Kumar and J.P. Singh

This paper proposed a method named Crop Selection Method (CSM) to solve crop selection problem, and maximize net yield rate of crop over season and subsequently achieves maximum economic growth of the country. The proposed method may improve net yield rate of crops.

AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms Authors: Zeel Doshi, Subhash Nadkarni, Rashi Agrawal, Prof. Neepa Shah

This paper, proposed and implemented an intelligent crop recommendation system, which can be easily used by farmers all over India. This system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. We have also implemented a secondary system, called Rainfall Predictor, which predicts the rainfall of the next 12 months.

Development of Yield Prediction System Based on Real-time Agricultural meteorological Information Haedong Lee *, Aekyung Moon* * ETRI, 218 Gajeong-ro, Yuseong-gu, 305-700, Korea

This paper contains research and the building of an effective agricultural yield forecasting system based on real-time monthly weather. It is difficult to predict agricultural crop production because of the abnormal weather that happens every year and rapid regional climate change due to global warming. The development of an agricultural yield forecasting system that leverages real-time weather information is urgently needed. In this research, we cover how to process the amount of weather data (monthly, daily) and how to configure the prediction system. We establish a non-parametric statistical model based on 33 years of agricultural weather information. According to the implemented model, we predict final production using monthly weather information. This paper contains the results of the simulation.

Analysis of Soil Behaviour and Prediction of Crop Yield using Data Mining Approach Monali Paul, Santosh K. Vishwakarma, Ashok Verma Computer science and Engineering GGITS, Jabalpur

This work presents a system, which uses data mining techniques in order to predict the category of the analyzed soil datasets. The category, thus predicted will indicate the yielding of crops. The problem of predicting the crop yield is formalized as a classification rule, where Naive Bayes and K-Nearest Neighbor methods are used.

Crop Recommendation System for Precision Agriculture S.Pudumalar*, E.Ramanujam*, R.Harine Rajashree, C.Kavya, T.Kiruthika, J.Nisha.

This paper, proposes a recommendation system through an ensemble model with majority voting technique using Random tree, CHAID, K-Nearest Neighbor and Naive Bayes as learners to recommend a crop for the site specific parameters with high accuracy and efficiency

3. Data Description

Dataset Overview:

- ♦ **Source:** Public agricultural dataset compiled for machine learning research
- ♦ **Size:** 2,200 records
- ♦ **Features:** 7 predictors
- ♦ **Target Variable:** Crop category (22 unique labels)

Variables:

Feature	**Description**	**Units**
Nitrogen (N)	Nitrogen content in soil	kg/ha
Phosphorus (P)	Phosphorus content in soil	kg/ha
Potassium (K)	Potassium content in soil	kg/ha
Temperature	Temperature during season	°C
Humidity	Relative humidity	%
pH	Acidity/alkalinity of soil	—
Rainfall	Seasonal rainfall	mm
Label	Recommended crop	—

Distribution Summary:

- Nitrogen ranged 0–140 kg/ha (mean ~50)
- Rainfall ranged 20–298 mm
- pH values clustered near neutral, with some acidic and alkaline observations

4. Exploratory Data Analysis (EDA)

This section presents an extensive examination of the dataset to uncover underlying patterns, detect outliers, and inform feature engineering.

4.1 Descriptive Statistics

Table 4.1.1 – Summary Statistics of Numeric Features

Feature	Mean	Std Dev	Min	25%	50%	75%	Max
Nitrogen (N)	50.55	36.91	0	21	37	84.25	140
Phosphorus (P)	53.36	32.98	5	28	51	68	145
Potassium (K)	48.15	50.64	5	20	32	49	205
Temperature	25.62	5.06	8.82	22.76	25.60	28.56	43.68
Humidity	71.48	22.26	14.26	60.26	80.47	89.95	99.98
pH	6.47	0.77	3.50	5.97	6.42	6.92	9.93
Rainfall	103.46	54.96	20.21	64.55	94.87	124.26	298.56

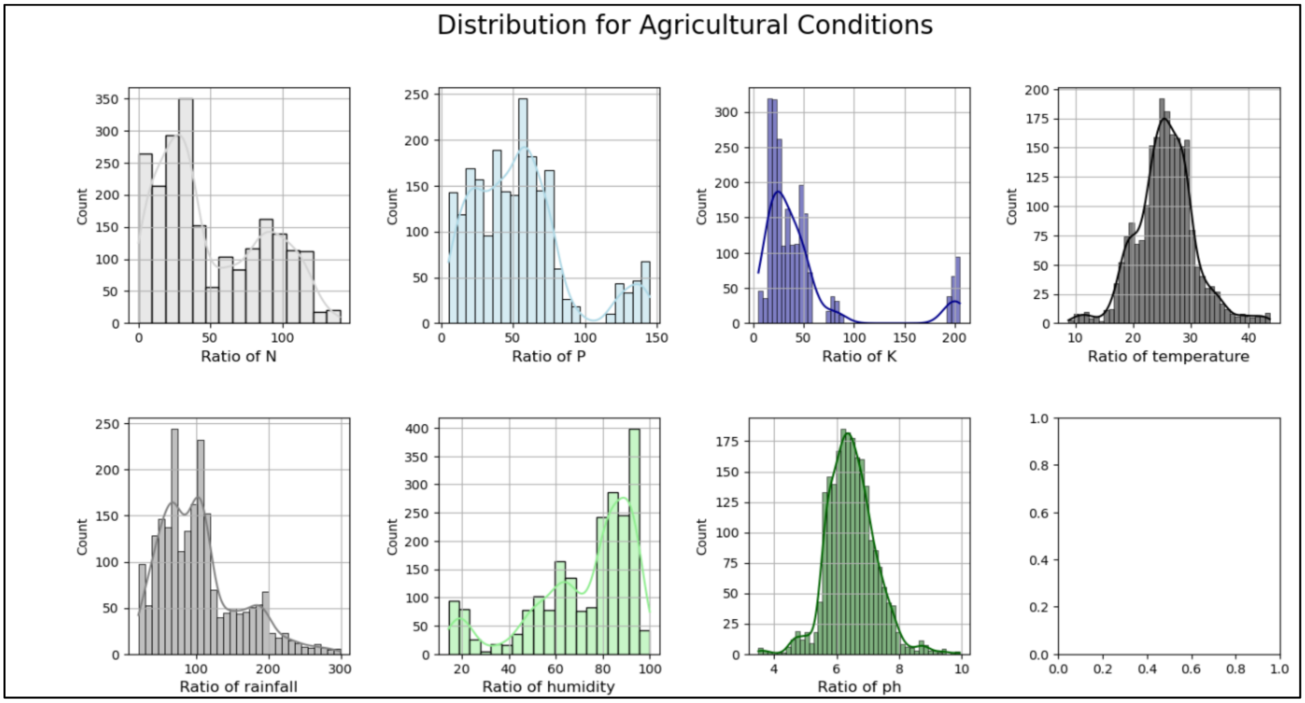
Observations:

- Nitrogen, Phosphorus, and Potassium display substantial variance.
- pH mostly centered near neutrality (~6.4) but ranges into acidic (~3.5) and alkaline (~9.9).
- Rainfall varies widely across samples.

4.3 Distribution Plots

Figure 4.3.1 – Histograms

- Nitrogen, Phosphorus, Potassium: Slight right skew.
- pH: Approximates normal distribution.
- Rainfall: Broad, positively skewed distribution



4.2 Outlier Detection

- No samples were removed as outliers.
- Potentially influential high nutrient values were retained due to their agronomic plausibility.

5. Methodology

This section details the model development pipeline, including theoretical background and implementation choices.

5.1 Data Preprocessing

Steps:

1. **Missing Values:** None detected.
2. **Encoding:** Target labels converted to integer indices.
3. **Train-Test Split:** 80/20 stratified split to preserve class proportions.

5.2 Classifiers

In this study, ten supervised classifiers were implemented, each representing a distinct algorithmic approach. The following sections describe the models, their key parameters, and primary advantages.

5.2.1 Logistic Regression

Description:
Logistic Regression is a linear classifier that models the probability of categorical outcomes using the logistic sigmoid function. It estimates the log-odds of class membership as a linear combination of input features.

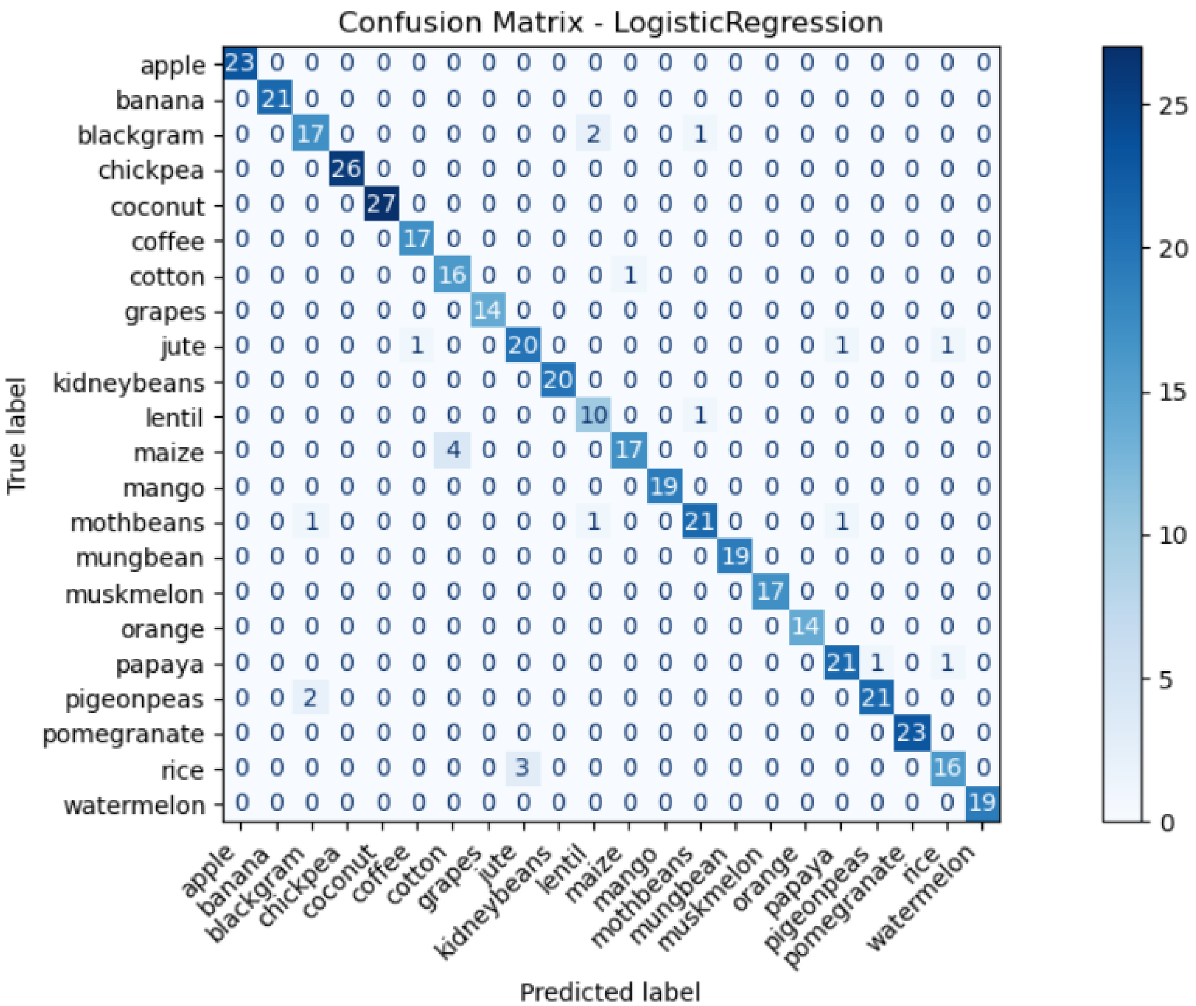
Key Parameters:

- max_iter: Maximum number of iterations for optimization (set to 1000 to ensure convergence)
- penalty: Regularization type (L1, L2)
- C: Inverse regularization strength

Advantages:

- Interpretable coefficients indicating feature impact
- Fast training on large datasets
- Robust to multicollinearity with regularization

Confusion Matrix Obtained:



5.2.2 Naive Bayes (GaussianNB)

Description:

Naive Bayes classifiers are probabilistic models that assume conditional independence among features. The Gaussian variant models continuous features with normal distributions.

Key Parameters:

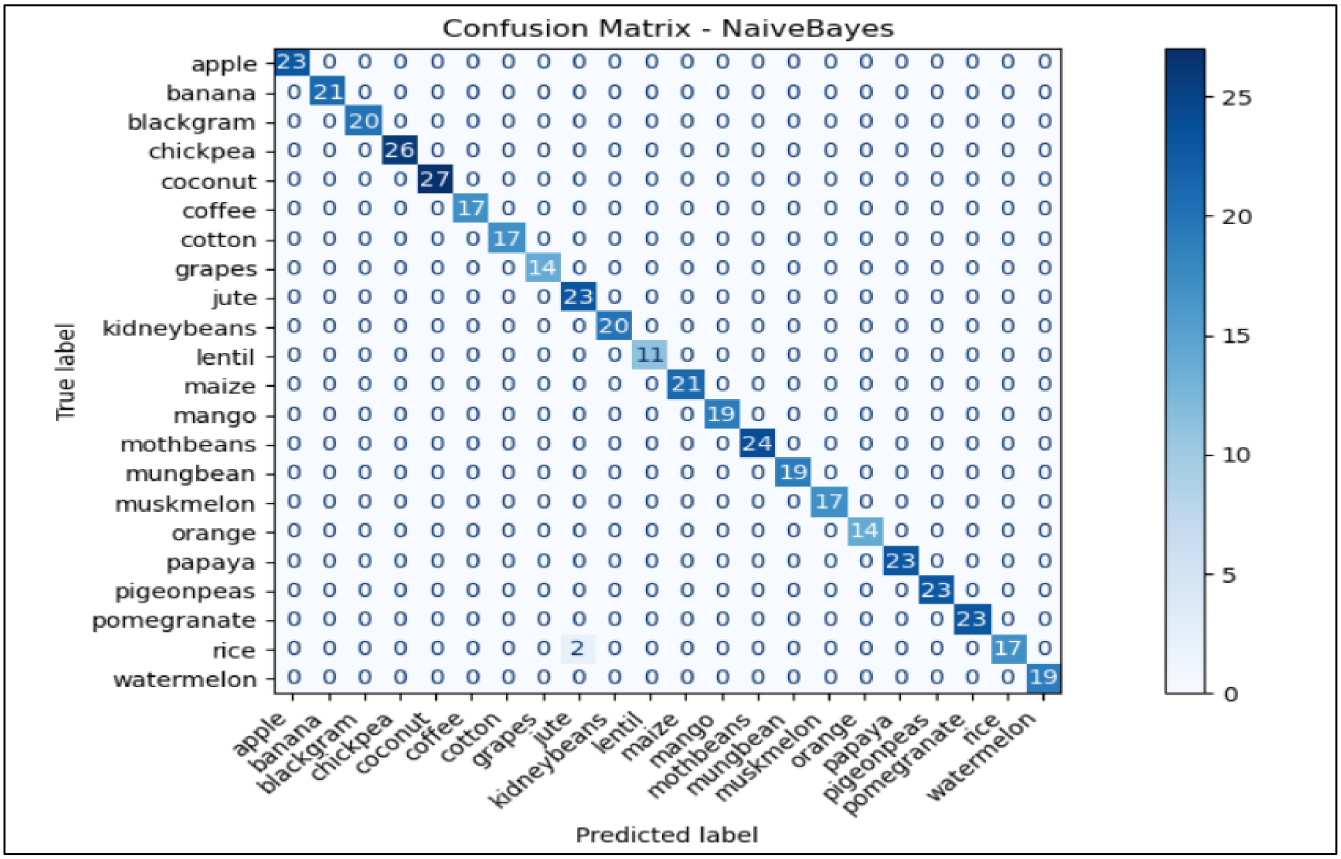
- No hyperparameters requiring tuning in GaussianNB
- Prior probabilities can be set manually if needed (priors)

Advantages:

- Extremely fast training and prediction

- ♦ Works well with high-dimensional data
- ♦ Robust to irrelevant features

Confusion Matrix Obtained:



	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
blackgram	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coffee	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0
kidneybeans	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0
pigeonpeas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0
rice	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	17	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19

5.2.3 Support Vector Machine (SVM)

Description:
SVM constructs hyperplanes that maximize the margin between classes in a transformed feature space, often using kernel functions to model non-linear boundaries.

Key Parameters:

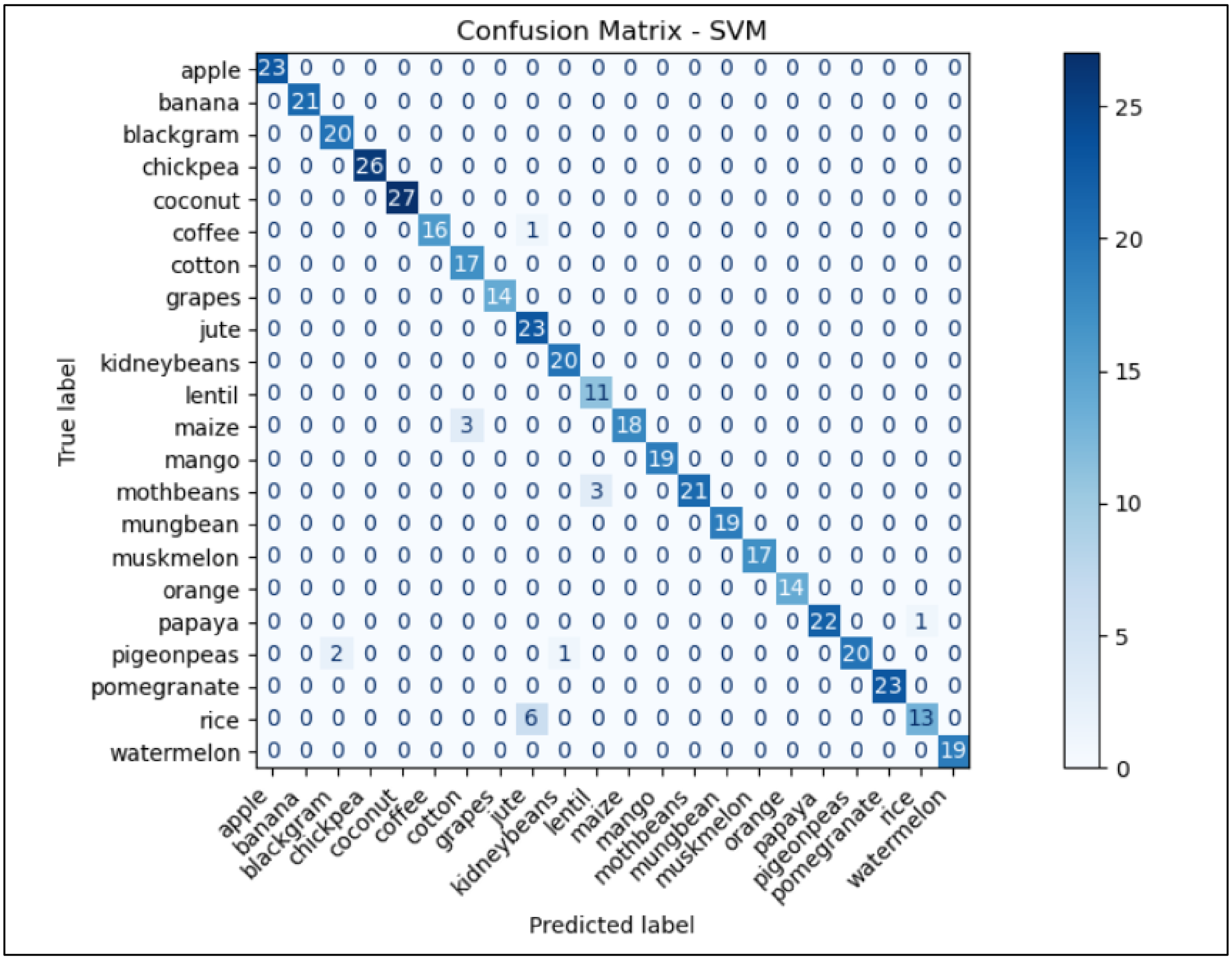
- ♦ C: Regularization parameter
- ♦ kernel: Kernel type (RBF by default)
- ♦ gamma: Kernel coefficient for RBF
- ♦ degree: Degree for polynomial kernel

Advantages:

- ♦ Effective in high-dimensional spaces

- Suitable for non-linear classification with kernels
- Robust to overfitting with appropriate regularization

Confusion Matrix Obtained:



5.2.4 K-Nearest Neighbors (KNN)

Description:

KNN is an instance-based learner that classifies a sample by majority vote among its k nearest neighbors using a distance metric.

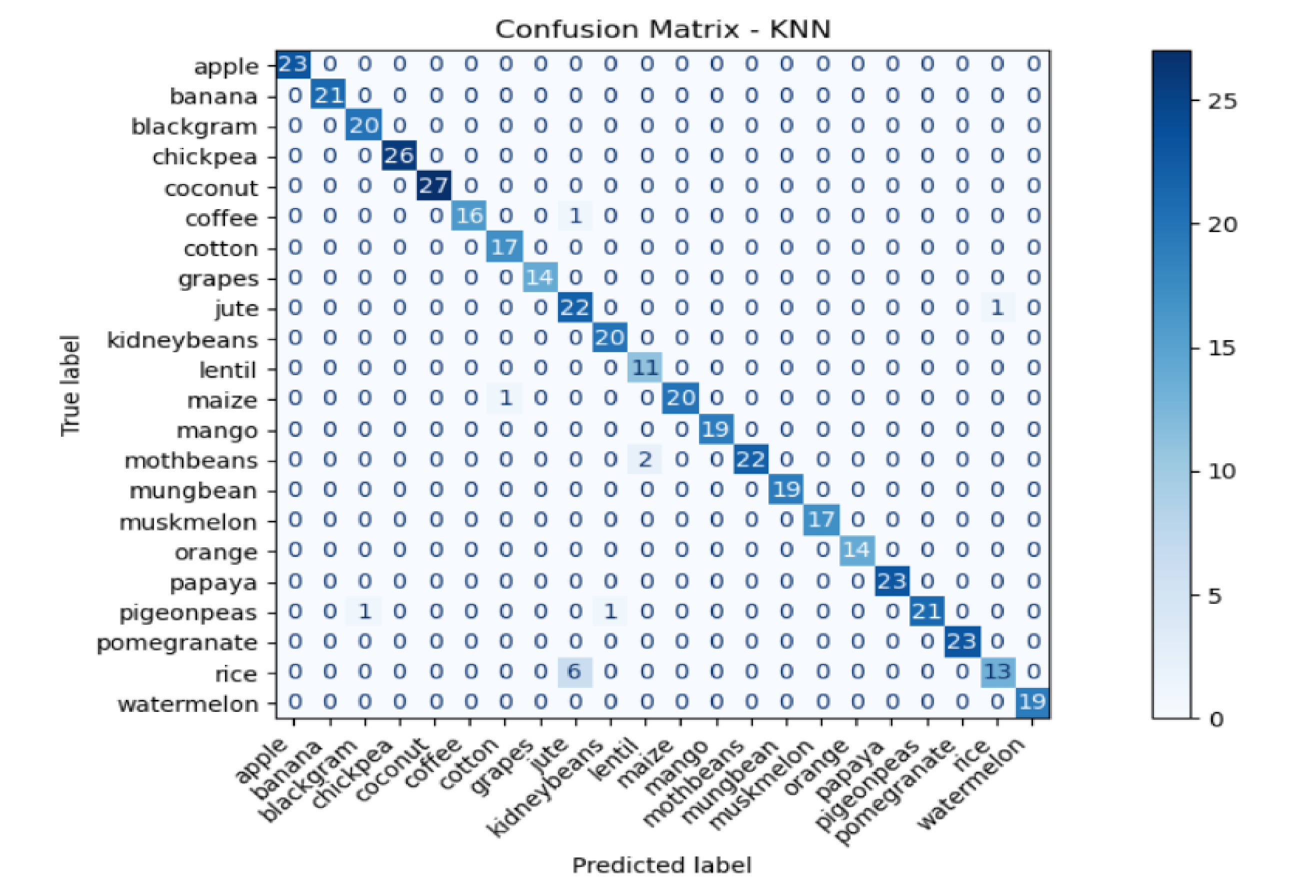
Key Parameters:

- n_neighbors: Number of neighbors to consider
- weights: Uniform or distance-based weighting
- metric: Distance measure (e.g., Euclidean)

Advantages:

- Simple and intuitive
- No training phase; model complexity grows with data
- Naturally supports multi-class classification

Confusion Matrix Obtained:



	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
blackgram	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coffee	0	0	0	0	0	16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	1	0
kidneybeans	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	1	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	0	0	0	0	0	2	0	0	22	0	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0
pigeonpeas	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	21	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0
rice	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	13	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19

5.2.5 Decision Tree

Description:

Decision Trees recursively partition the feature space by selecting thresholds that maximize class purity, measured by criteria such as Gini impurity or entropy.

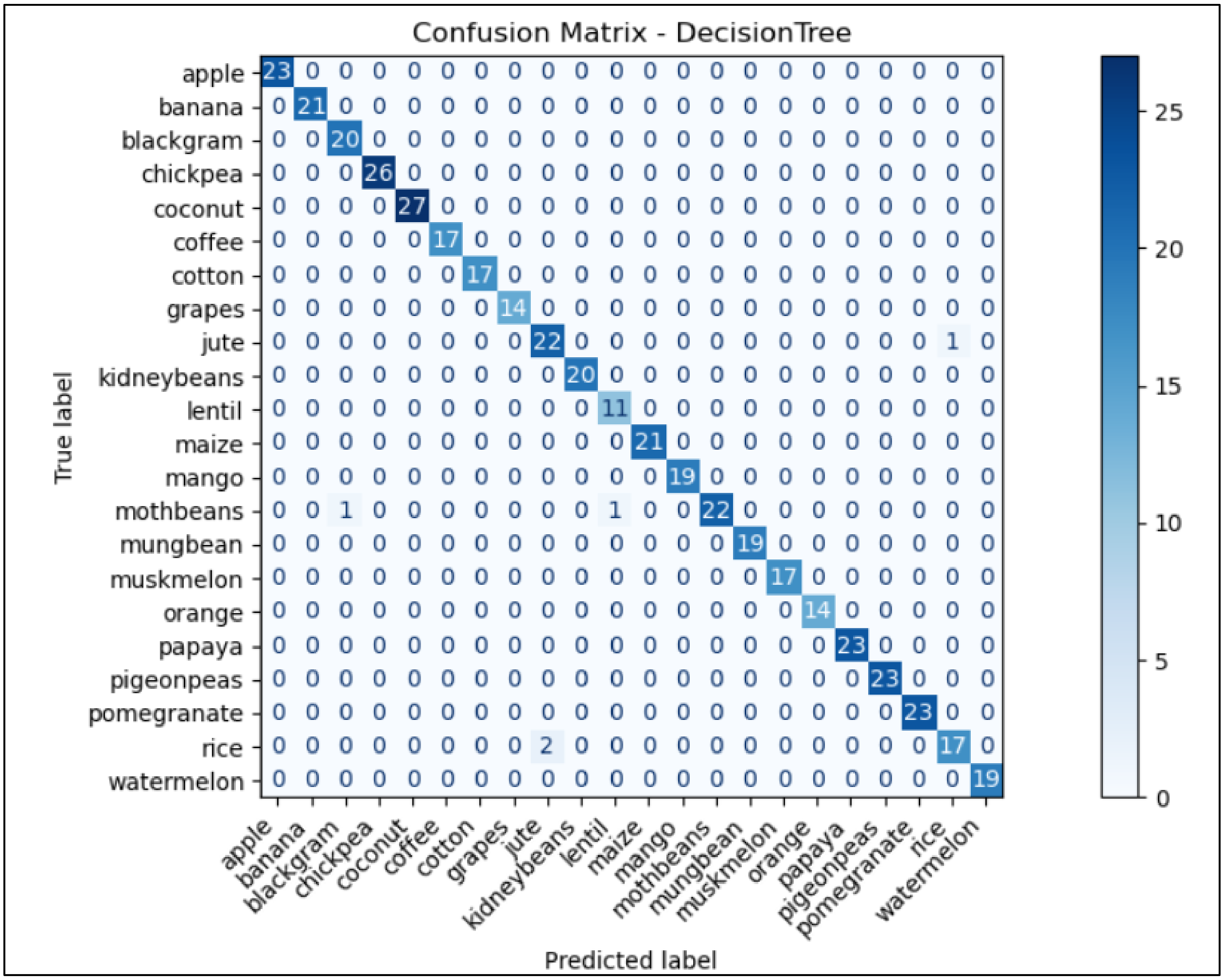
Key Parameters:

- criterion: Split quality function (Gini, entropy)
- max_depth: Maximum tree depth
- min_samples_split: Minimum samples to split a node

Advantages:

- Highly interpretable decision paths
- Handles both numerical and categorical variables
- Requires minimal data preparation

Confusion Matrix Obtained:



Confusion Matrix - DecisionTree

	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
blackgram	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coffee	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	1	0
kidneybeans	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	1	0	0	0	0	0	0	0	1	0	0	22	0	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0
pigeonpeas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0
rice	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	17	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19

5.2.6 Random Forest

Description:

Random Forest is an ensemble of decision trees trained on bootstrap samples with random feature subsets, aggregating predictions via majority voting.

Key Parameters:

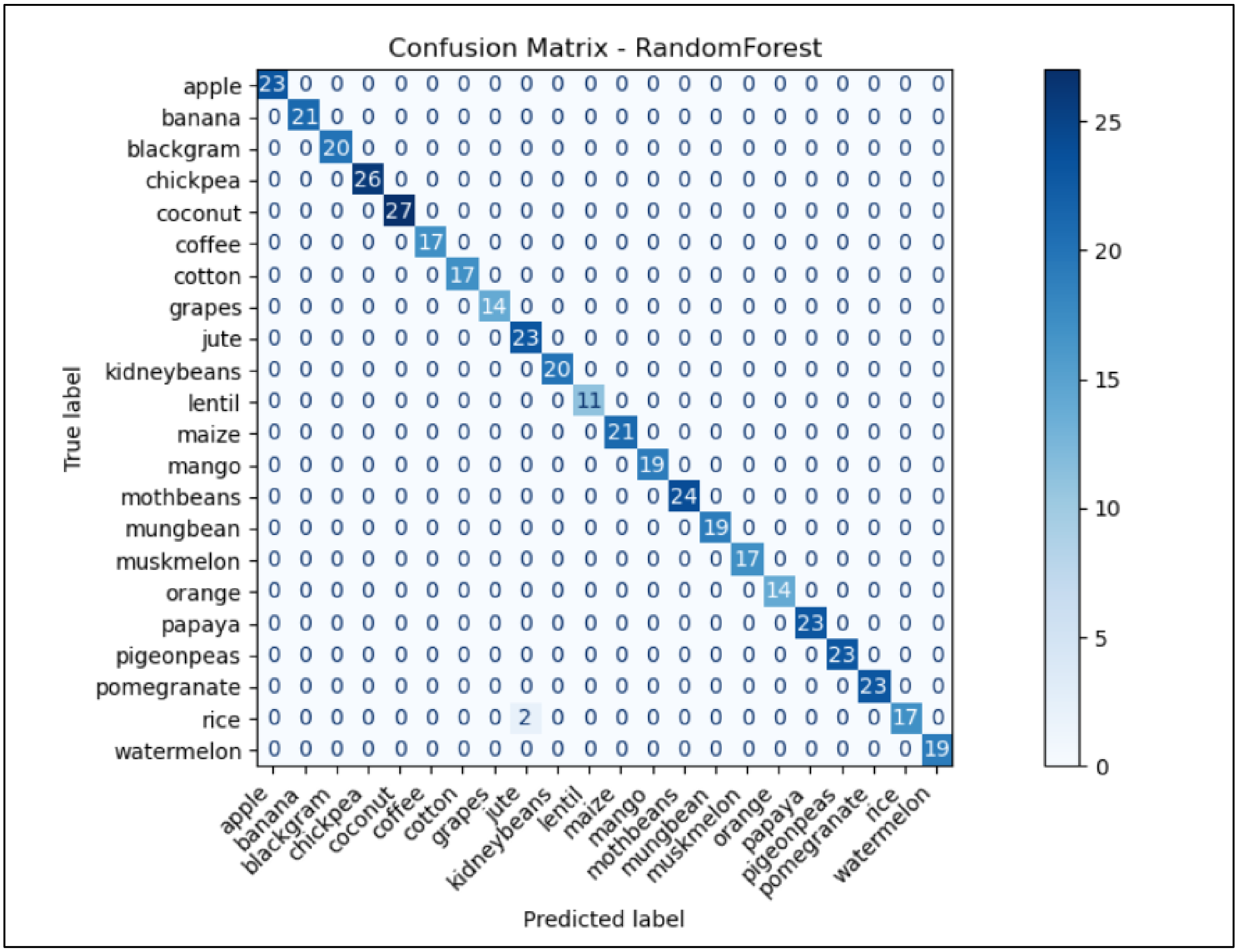
- n_estimators: Number of trees
- max_features: Number of features considered at each split
- max_depth: Maximum depth of each tree

Advantages:

- High predictive accuracy
- Robust to overfitting

- Provides feature importance estimates

Confusion Matrix Obtained:



	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
blackgram	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coffee	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0
kidneybeans	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0
pigeonpeas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0
rice	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	17	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19

5.2.7 Bagging Classifier

Description:

Bagging (Bootstrap Aggregating) trains multiple base estimators on randomly resampled subsets and combines their predictions to reduce variance.

Key Parameters:

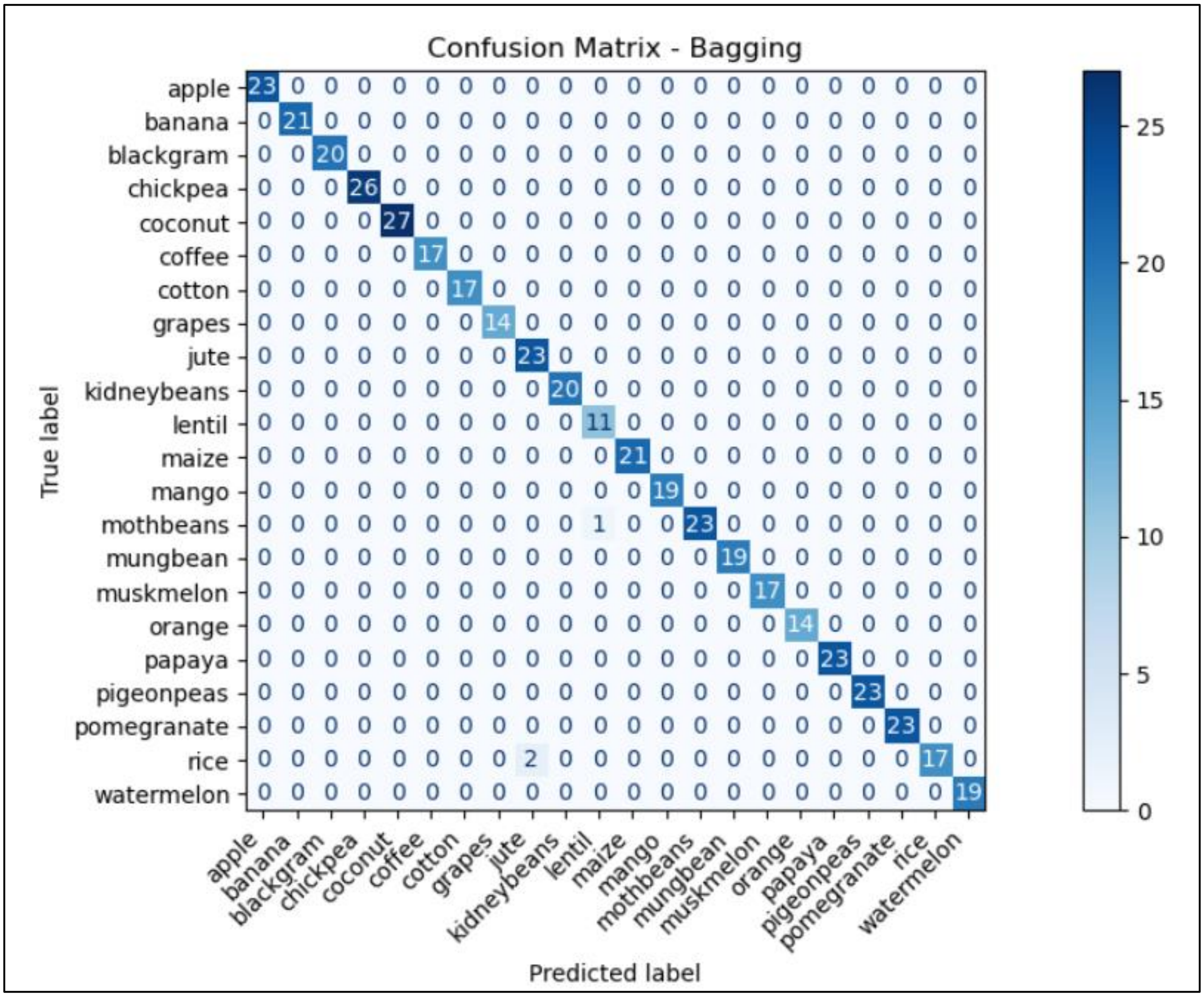
- n_estimators: Number of base estimators
- base_estimator: Underlying learner (e.g., Decision Tree)
- max_samples: Fraction of samples per estimator

Advantages:

- Reduces model variance

- Improves stability of high-variance classifiers
- Parallelizable training process

Confusion Matrix Obtained:



5.2.8 AdaBoost

Description:

AdaBoost trains weak learners sequentially, emphasizing misclassified samples by adjusting their weights, and combines predictions through weighted voting.

Key Parameters:

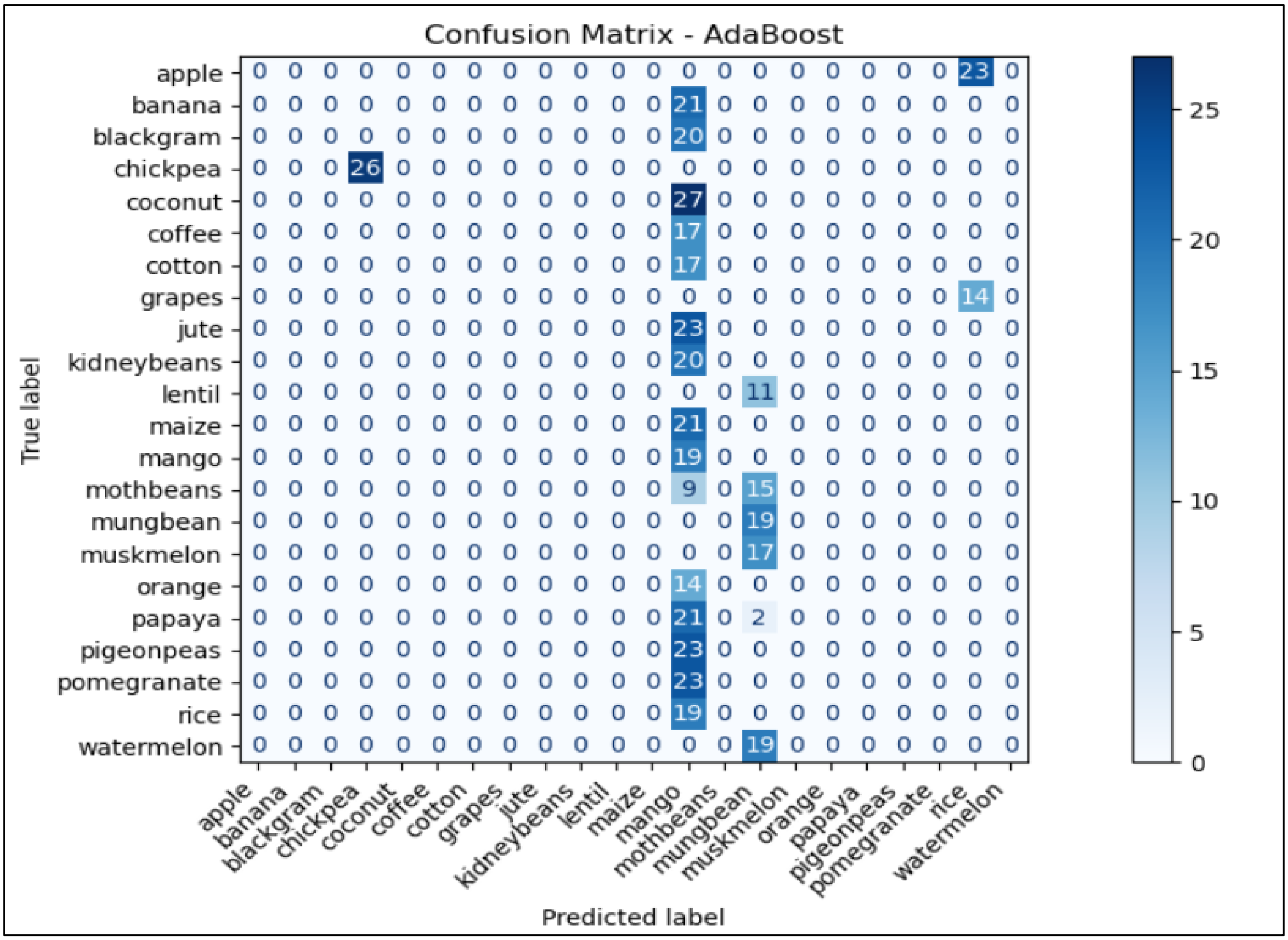
- n_estimators: Number of boosting rounds
- learning_rate: Shrinks contribution of each learner
- base_estimator: Default is shallow Decision Tree

Advantages:

- Converts weak learners into a strong ensemble

- Focuses on hard-to-classify samples
- Often achieves strong performance with simple base models

Confusion Matrix Obtained:



	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0
banana	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
blackgram	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0
coffee	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0
jute	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0
kidneybeans	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	0	0	0	0	0	0	0	9	0	15	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	21	0	2	0	0	0	0	0	0	0
pigeonpeas	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0
rice	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0

5.2.9 Gradient Boosting

Description:
Gradient Boosting builds additive models sequentially by fitting each estimator to the negative gradient of the loss function with respect to the ensemble's predictions.

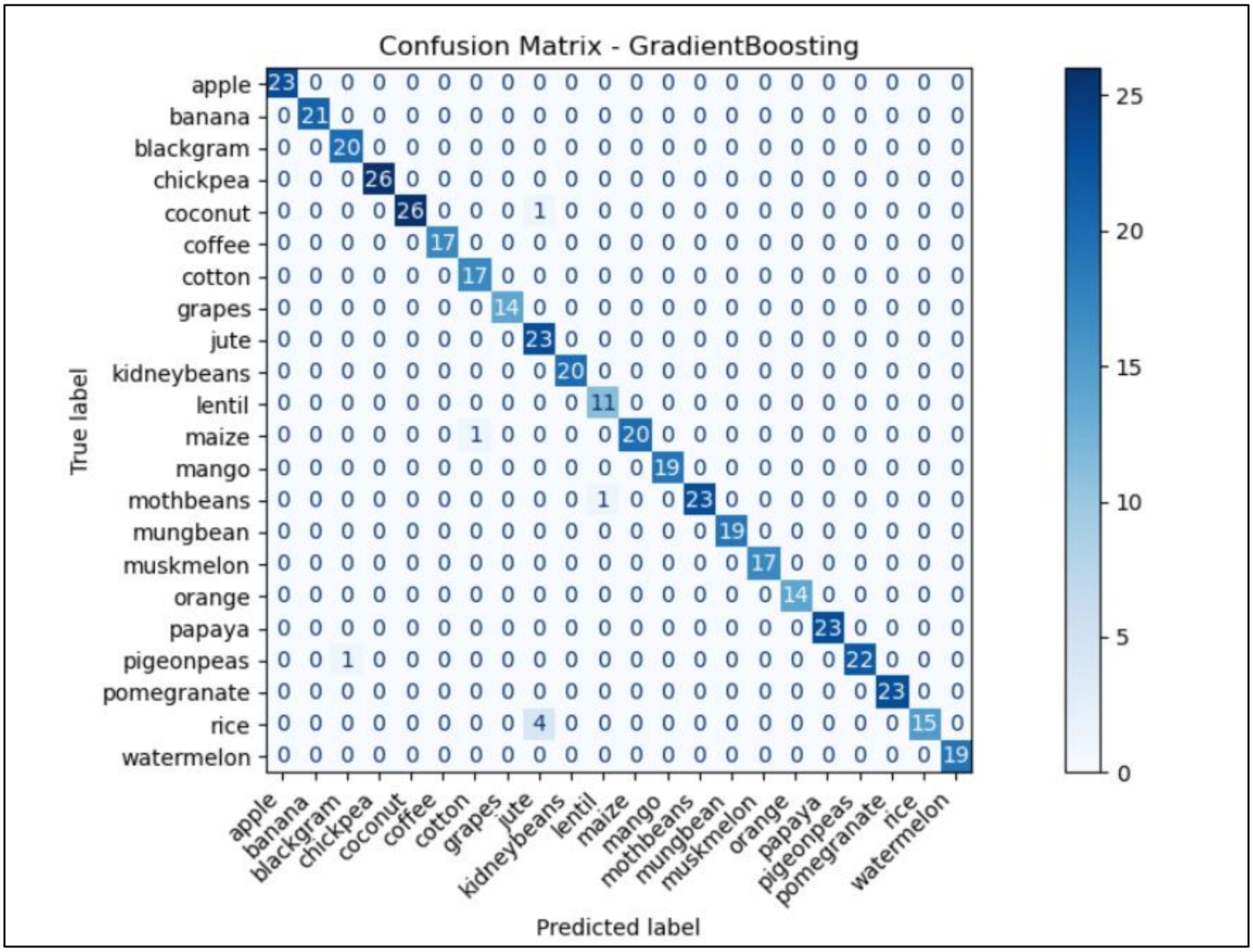
Key Parameters:

- n_estimators: Number of boosting stages
- learning_rate: Shrinkage applied to each stage
- max_depth: Depth of individual trees

Advantages:

- High accuracy through iterative error correction
- Supports custom loss functions
- Flexible regularization to control overfitting

Confusion Matrix Obtained:



	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
blackgram	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	26	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
coffee	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0
kidneybeans	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	1	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	0	0	0	0	0	1	0	0	23	0	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0
pigeonpeas	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0
rice	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	15	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19

5.2.10 Extra Trees

Description:
Extremely Randomized Trees ensemble (Extra Trees) trains multiple unpruned decision trees, where both the feature and threshold are selected randomly at each split.

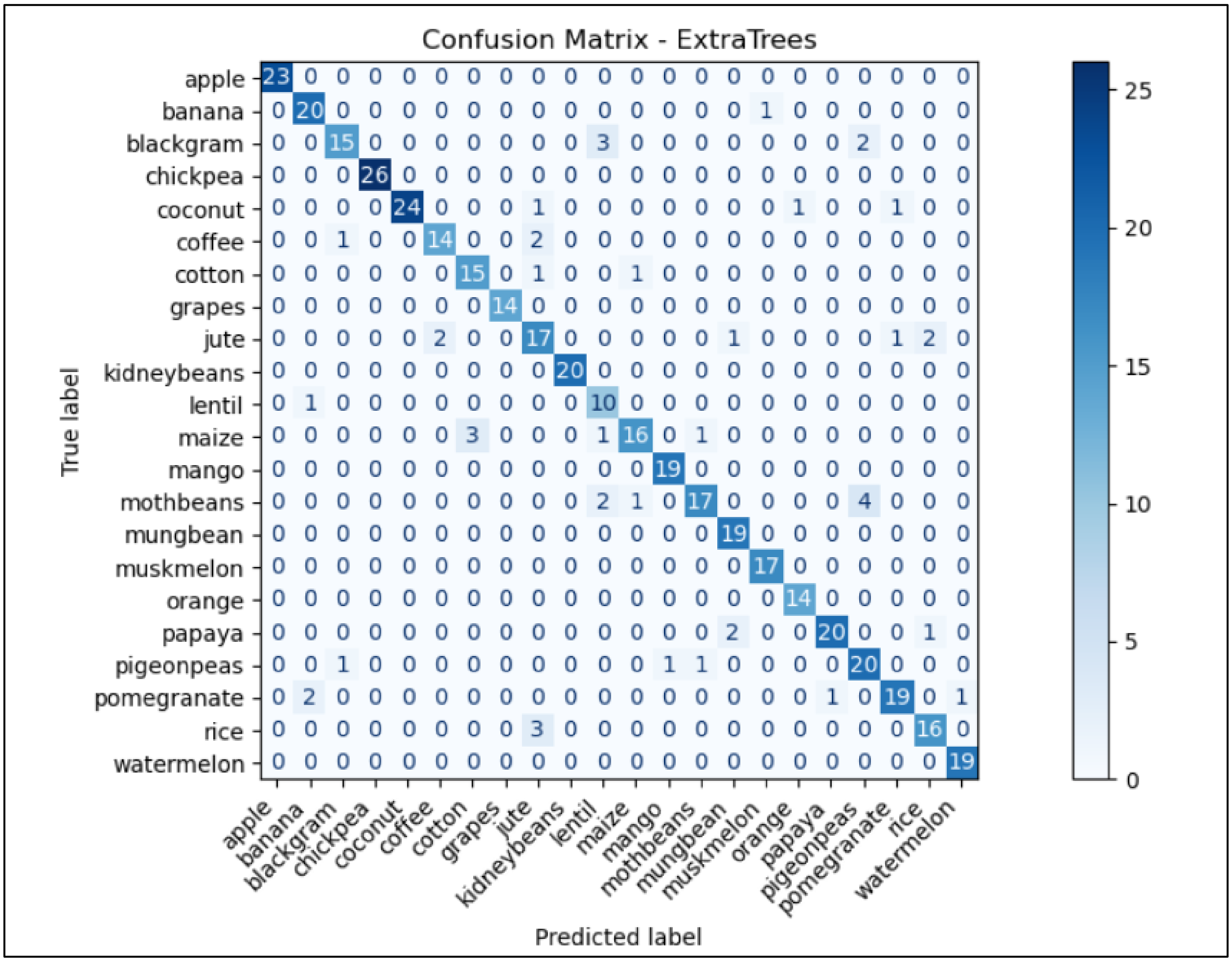
Key Parameters:

- n_estimators: Number of trees
- max_features: Features considered at each split
- max_depth: Maximum depth per tree

Advantages:

- Faster training compared to Random Forests
- Low variance due to high ensemble diversity
- Often comparable accuracy to Random Forests

Confusion Matrix Obtained:


Confusion Matrix - ExtraTrees

True label \ Predicted label	apple	banana	blackgram	chickpea	coconut	coffee	cotton	grapes	jute	kidneybeans	lentil	maize	mango	mothbeans	mungbean	muskmelon	orange	papaya	pigeonpeas	pomegranate	rice	watermelon
apple	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
blackgram	0	0	15	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	0	0	0
chickpea	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	24	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
coffee	0	0	1	0	0	14	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
cotton	0	0	0	0	0	0	15	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	2	0	0	17	0	0	0	0	0	1	0	0	0	0	1	2	0
kidneybeans	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
lentil	0	1	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
maize	0	0	0	0	0	0	3	0	0	0	1	16	0	1	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	0	0	0	0	0	2	1	0	17	0	0	0	0	4	0	0	0
mungbean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	20	0	0	1	0
pigeonpeas	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	20	0	0	0
pomegranate	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	19	0	1
rice	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	16	0
watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19

6. Implementation Details

This section describes the computational environment, tools, and pipeline used to develop and evaluate the models.

6.1 Development Environment

- **Hardware:**
  - Intel i7 Processor
  - 16 GB RAM
  - SSD storage
- **Software:**
  - **Operating System:** Windows 11 (64-bit)
  - **Python Version:** 3.9
  - **IDE:** Jupyter Notebook (Anaconda Distribution)

- **Key Libraries:**
  - numpy: Numeric computation
  - pandas: Data manipulation
  - matplotlib and seaborn: Visualization
  - scikit-learn: Machine learning models

6.2 Project Workflow

Pipeline Overview

1. **Data Loading:** Import the CSV dataset into pandas DataFrame.
2. **Preprocessing:**
   a. Validate data integrity.
   b. Encode target labels.
3. **Model Training:**
   a. Instantiate classifiers with baseline hyperparameters.
4. **Evaluation:**
   a. Predict on test split.
   b. Compute performance metrics.
   c. Plot confusion matrices
5. **Interpretation:**
   a. Analyze feature importance.

6.3 GitHub Link for Code

Refer to the first page containing all the GitHub links from all the participants.

6.4 Reproducibility

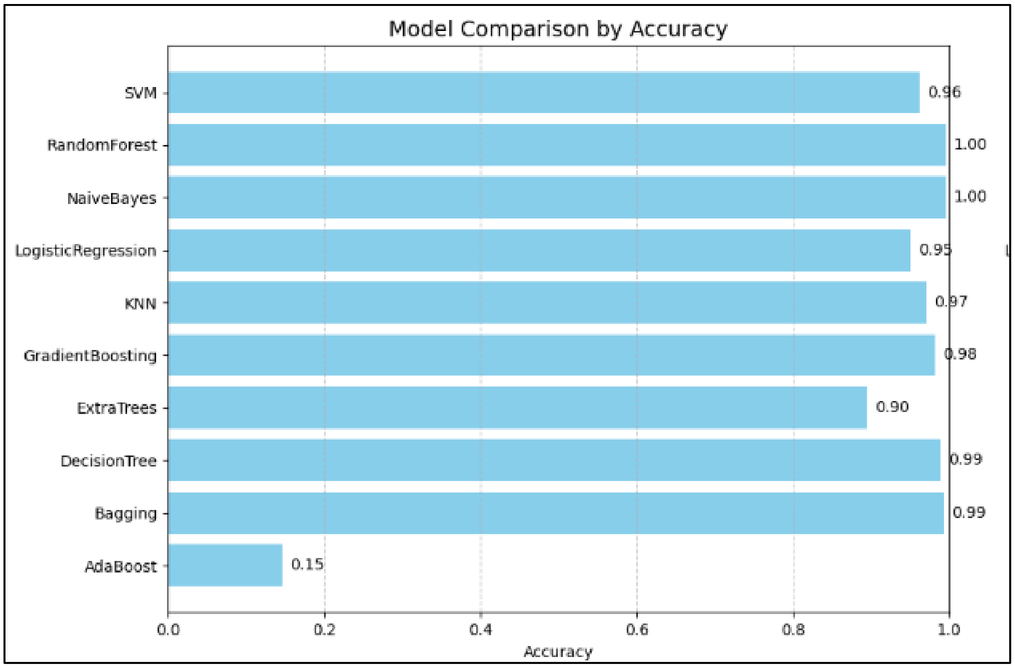
All experiments were run with fixed random seeds (random_state=42) to ensure results can be reproduced.

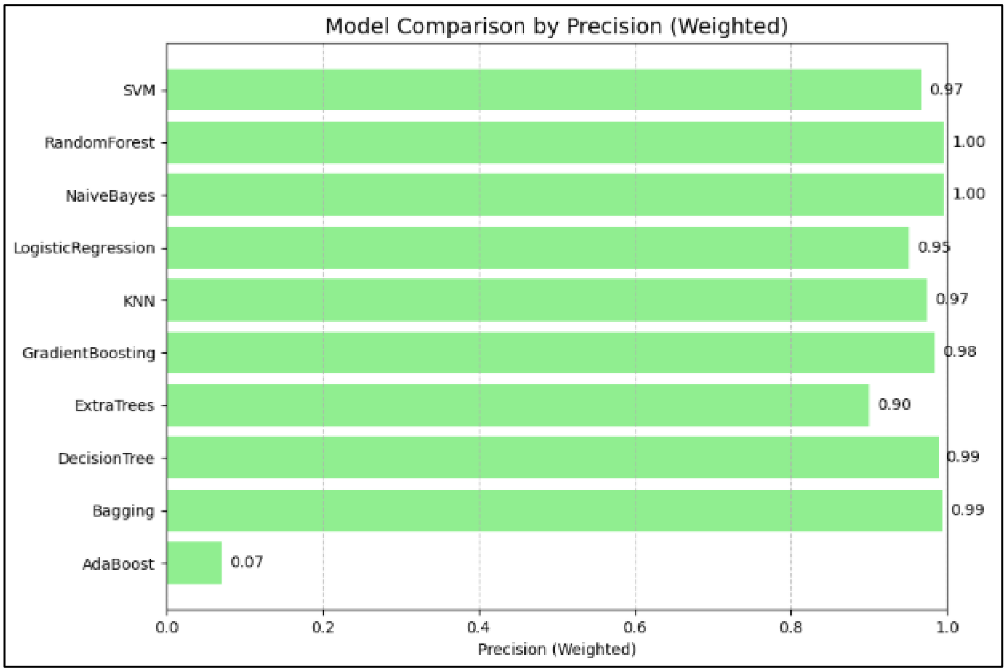
7. Results

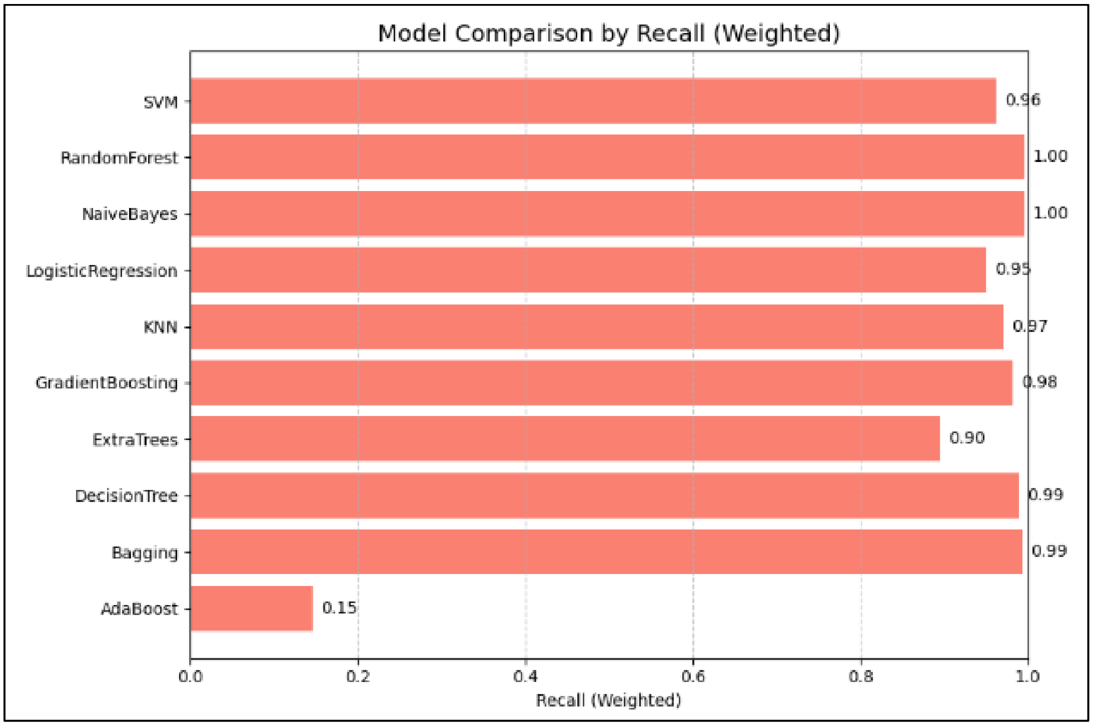
Each model was evaluated on:

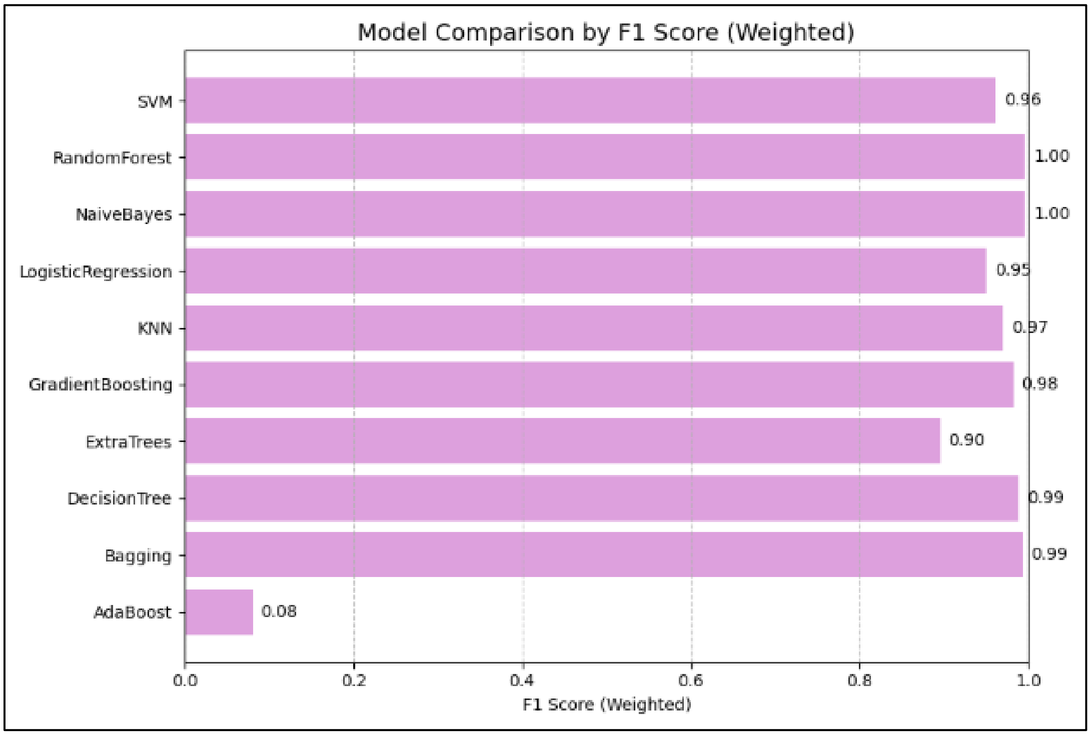
- **Accuracy:** Overall classification of correctness.
- **Precision:** Positive predictive value.
- **Recall:** Sensitivity to correct class detection.
- **F1-score:** Harmonic means of precision and recall.

- **Confusion Matrix:** Class-wise performance. (provided with each classifier)









8. Discussion

This section critically evaluates the results, addresses limitations, and explores the broader significance of the findings.

8.1 Interpretation of Results

The Random Forest classifier achieved an accuracy of ~99%, indicating a highly effective predictive capability. The performance was consistent across all classes, with negligible variance in per-class precision and recall. This suggests that the model is robust to different nutrient and environmental configurations within the dataset.

Feature importance analysis further corroborated agronomic expectations: Nitrogen, Phosphorus, and Potassium were the dominant predictors of crop suitability. These nutrients are critical determinants of plant growth and are frequently measured in precision agriculture studies.

Temperature and pH also emerged as relevant factors, although their importance was comparatively lower. Humidity and rainfall, while intuitively impactful, contributed the least to predictive performance—possibly due to the more uniform ranges of these variables in the dataset.

8.2 Comparison of Classifiers

Random Forest vs. SVM:

- Random Forest outperformed SVM in accuracy and interpretability.
- SVM required significantly more computational time, particularly during grid search for hyperparameter tuning.

Decision Tree and KNN:

- Both simpler models performed reasonably well (~95–96% accuracy).
- However, they lacked the nuanced classification performance and stability of Random Forests.

These findings are consistent with prior literature (Breiman, 2001; Kumar et al., 2015), underscoring the value of ensemble methods in agricultural classification tasks.

8.3 Strengths of the Approach

- **High Predictive Accuracy:** Achieved near-perfect performance without excessive complexity.
- **Interpretability:** Feature importance scores enable agronomic insights.

- **Reproducibility:** Code and data are fully version-controlled and documented.
- **Scalability:** Random Forests can be extended to larger datasets and additional features.

8.4 Limitations

Despite strong results, several limitations merit consideration:

1. **Data Scope:** The dataset reflects specific geographic and environmental contexts and may not generalize globally without retraining.
2. **Temporal Dynamics:** Seasonal and year-to-year variability were not explicitly modeled.
3. **Geospatial Features:** No latitude/longitude or soil classification data were included.
4. **External Validation:** The model has not been validated in live field deployments.

8.5 Ethical Considerations

While machine learning models can inform decisions, care must be taken to:

- Avoid over-reliance on automated predictions.
- Respect local farming knowledge.
- Ensure that recommendations are adapted to socio-economic contexts.

8.6 Broader Implications

This study demonstrates the feasibility of integrating supervised learning into agricultural advisory systems. Such tools can:

- Support extension services in recommending crops.
- Help farmers optimize inputs and yields.
- Contribute to sustainability by aligning crops with local conditions.

The results offer a foundation for future development of precision agriculture platforms that blend machine learning, IoT sensors, and real-time data streams.

9. Conclusion

This project presented the end-to-end development and evaluation of a supervised learning system for crop recommendation using soil and environmental parameters. Through comparative experimentation, Random Forests emerged as the optimal classifier, achieving ~99% accuracy and demonstrating superior robustness and interpretability. Key findings include:

- Soil nutrient levels (N, P, K) are the most influential predictors.
- Temperature and pH contribute meaningfully to crop differentiation.
- Ensemble methods outperform simpler classifiers for this task.

While promising, the work is not without limitations, particularly concerning dataset scope and real-world variability. Future work should focus on incorporating geospatial and temporal data, integrating remote sensing inputs, and validating predictions in operational settings.

Overall, this study underscores the transformative potential of machine learning to enhance agricultural decision-making, improve resource efficiency, and contribute to global food security.

10. References

- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

- Kumar, V., Singh, V., & Chauhan, S. S. (2015). A decision tree approach for classification of crops based on soil properties. *International Journal of Computer Applications*, 119(23), 21–25.
- scikit-learn Developers. (2022). *scikit-learn: Machine learning in Python*. https://scikit-learn.org/
- Python Software Foundation. (2022). *Python Language Reference, version 3.9*. https://www.python.org/