Smart Crop Recommendation using Machine Learning

“AI powered crop prediction based on soil and weather data”

1. Project Definition and Societal Context

1.1. Project Objective

The objective of this project is to build a **machine learning-based Crop Recommendation System** that suggests the most suitable crop for a given farmland based on environmental and soil parameters. The model uses features such as:

- Nitrogen (N)
- Phosphorus (P)
- Potassium (K)
- Temperature
- Humidity
- pH
- Rainfall

This system helps farmers and agricultural departments by:

- Improving farming decisions using data-driven insights
- Preventing crop failure due to soil–crop mismatch
- Increasing productivity and revenue
- Reducing the misuse of fertilizers and resources
- Supporting precision agriculture and sustainable farming practices

Overall, this project builds an intelligent crop advisory system that enhances agricultural efficiency and promotes smarter cultivation choices.

1.2. Alignment with UN Sustainable Development Goals (SDG)

- **SDG 2 – Zero Hunger :**
  Helps increase agricultural productivity through scientific crop selection.
  Reduces crop failure and promotes food security.

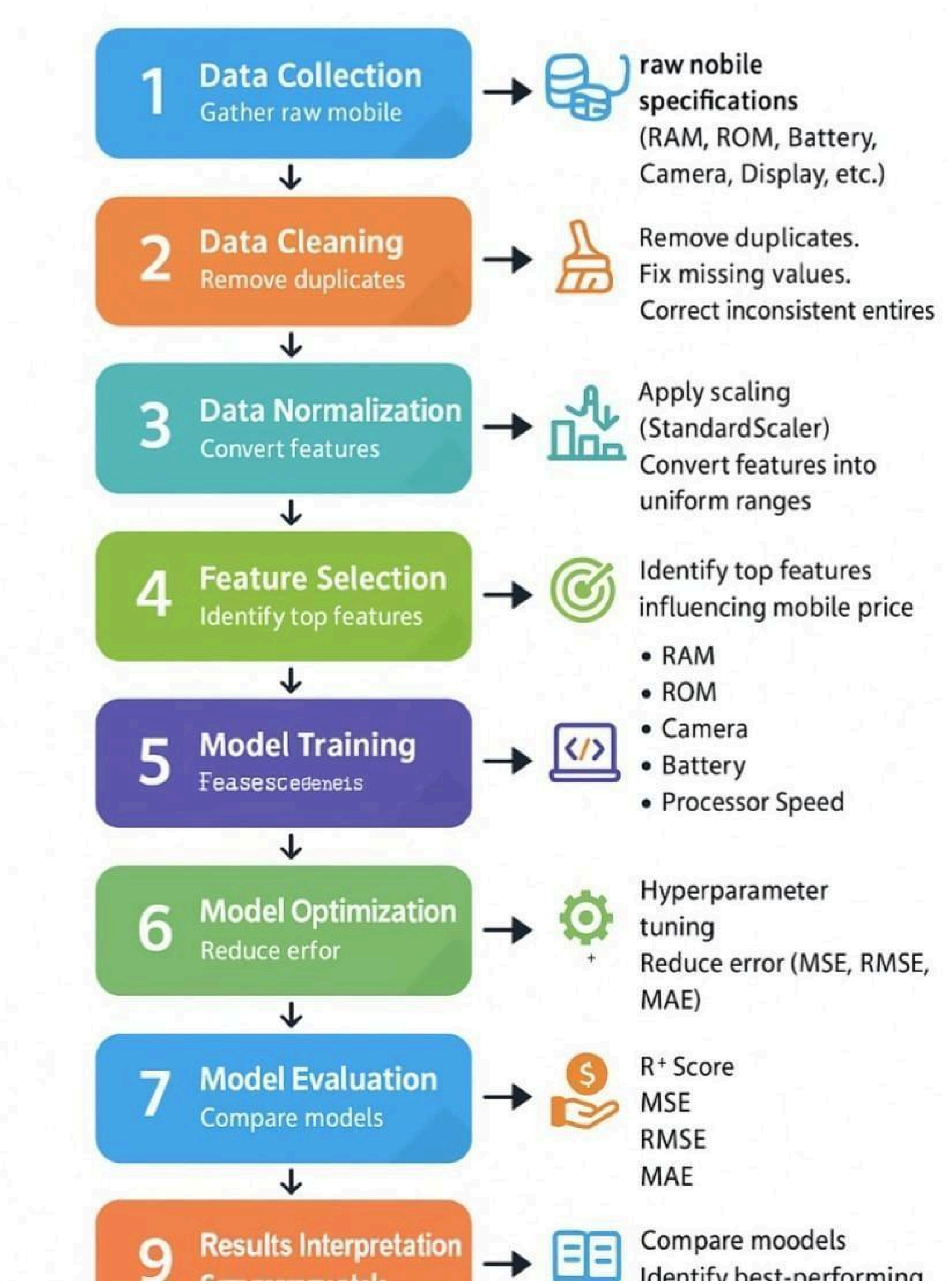
- **SDG 12 – Responsible Consumption & Production :**
  Prevents excessive fertilizer use
  Promotes sustainable crop planning based on soil capability.
- **SDG 13 – Climate Action :**
  Encourages climate-resilient farming using temperature–rainfall-based decisions.
- **SDG 9 – Industry, Innovation & Infrastructure :**
  Incorporates ML-based systems in agriculture
  Supports digital transformation in farming.
- **SDG 4 – Quality Education :**
  Enhances practical understanding of AI applications in agriculture**.**

2 ZERO HUNGER
9 INDUSTRY INNOVA AND INFRASTRUCTURE
13 RESPONSIBLE CONSUMPTION AND PRODUCTION
4 QUALITY EDUCATION

1.3. Literature Review

Author & Year	Methods Used	Relevance to Present Project
Patel, S. et al. (2024)	Random Forest, SVM, KNN	Evaluates the efficiency of ML models in crop prediction using soil nutrients and climate. Inspires use of ensemble methods.
Rao, A. & Deshmukh, P. (2025)	Decision Tree, Naive Bayes	Highlights importance of simple interpretable models for agriculture advisory systems.
Li, Z. (2024)	Gradient Boosting Models	Shows boosting algorithms provide high accuracy in agriculture datasets
Mohammed, Y. (2025)	SVM, Logistic Regression	Demonstrates the usefulness of SVM in multi-class classification problems like crop recommendation.
Farooq, I. (2024)	KNN, Random Forest	Reinforces that RF and KNN perform very well on soil–environment datasets.

1.4. BLOCK DIAGRAM



1 Data Collection
Gather raw mobile

raw nobile specifications
(RAM, ROM, Battery, Camera, Display, etc.)

2 Data Cleaning
Remove duplicates

Remove duplicates.
Fix missing values.
Correct inconsistent entires

3 Data Normalization
Convert features

Apply scaling (StandardScaler)
Convert features into uniform ranges

4 Feature Selection
Identify top features

Identify top features influencing mobile price

- RAM
- ROM
- Camera
- Battery
- Processor Speed

5 Model Training
Feasescedeneis

6 Model Optimization
Reduce erfor

Hyperparameter tuning
Reduce error (MSE, RMSE, MAE)

7 Model Evaluation
Compare models

R⁺ Score
MSE
RMSE
MAE

9 Results Interpretation

Compare moodels
Identify best-performing

2. Data Understanding and Exploratory Data Analysis (EDA)

2.1. Data Loading and Initial Assessment

The dataset contains 2200 real agricultural records.Each row includes soil nutrients, climate conditions, and crop label.

- **Target Variable:**

label (A categorical variable representing the recommended crop such as rice, maize, mango, banana, orange, apple, etc.)

- **Main Numerical Features:**

➔ N (Nitrogen content in soil)

➔ P (Phosphorus content)

➔ K (Potassium content)

➔ temperature (°C)

➔ humidity (%)

➔ ph (acidity/alkalinity of soil)

➔ rainfall (mm)

These continuous features directly determine the suitability of different crops based on soil chemistry and climate conditions.

- **Categorical Features:**

Label (Crop Type)
Already stored as text values and later encoded using LabelEncoder, producing numerical class labels.

- **Handling Missing Values:**

➔ The dataset contains no missing values, making preprocessing straightforward.

➔ All independent variables are numerical, and the target variable was encoded before modeling.

➔ Outlier inspection showed the data to be clean and well-distributed across feature ranges.

2.2. Target Variable Analysis: The Core Challenge

Crop recommendation is a multi-class classification problem, as the model needs to choose the most suitable crop out of 22 possible crop categories.

Observed Patterns

- Crops like rice require high rainfall and high humidity.
- Crops like chickpea and kidneybeans prefer low rainfall and moderate nutrient levels.
- High Nitrogen (N) often corresponds to leafy or nutrient-demanding crops.
- The relationship between soil nutrients, climate conditions, and crops is nonlinear, making ML classification techniques suitable for accurate prediction.

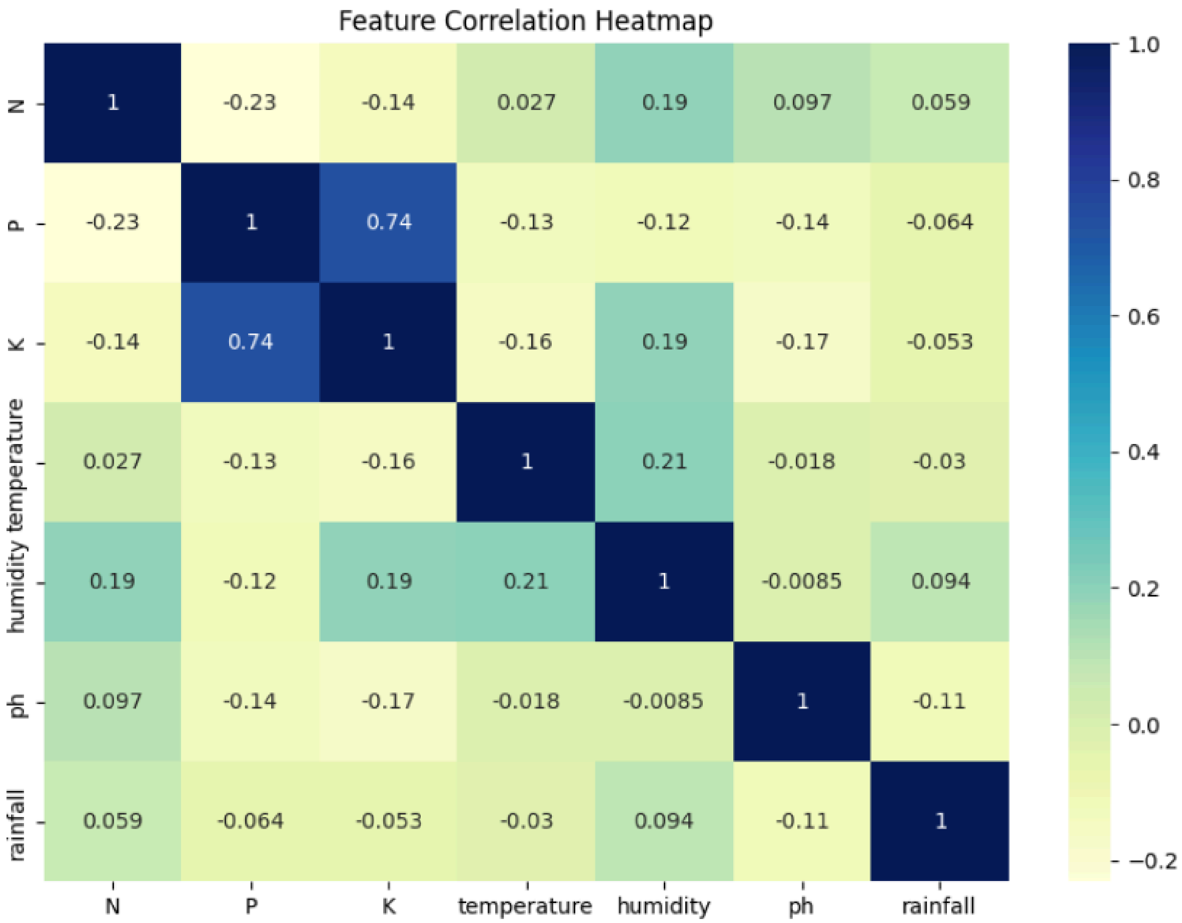
Evaluation Metrics Used :

- Accuracy Score
- Precision
- Recall
- F1-Score
- Confusion Matrix

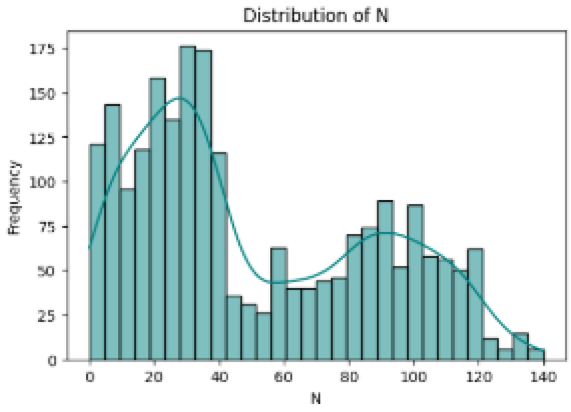
2.3.2 Relationship Visualization

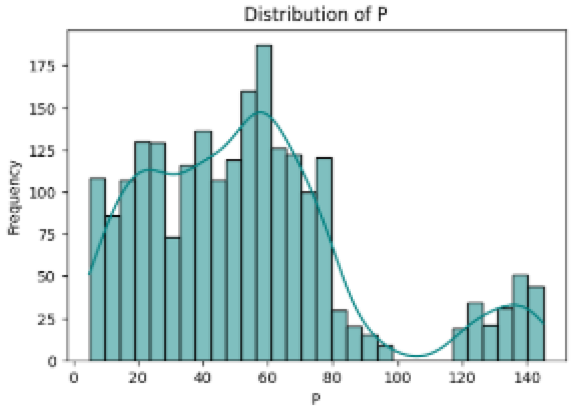
1.Correlation Heatmap

...

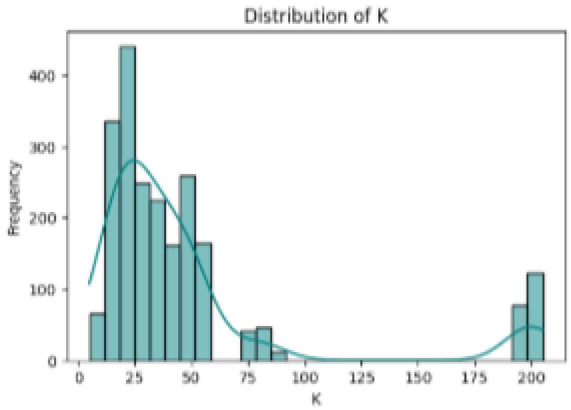


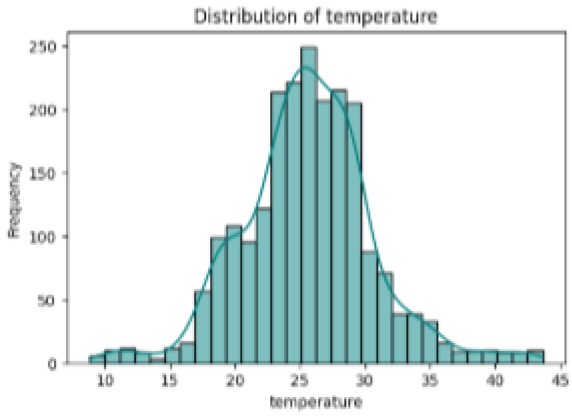
2.Distribution of each feature

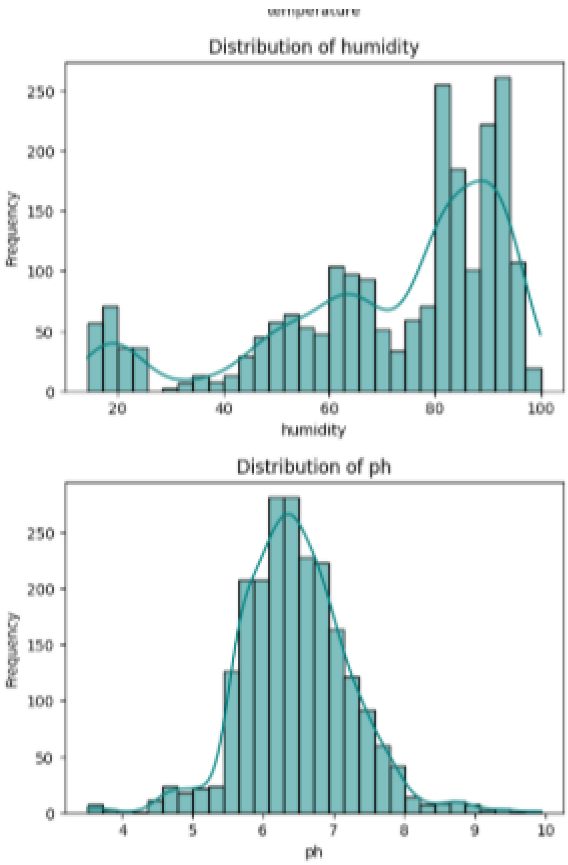

Distribution of N
Frequency
0
25
50
75
100
125
150
175
0
20
40
60
80
100
120
140
N


Distribution of P
Frequency
0
25
50
75
100
125
150
175
0
20
40
60
80
100
120
140
P

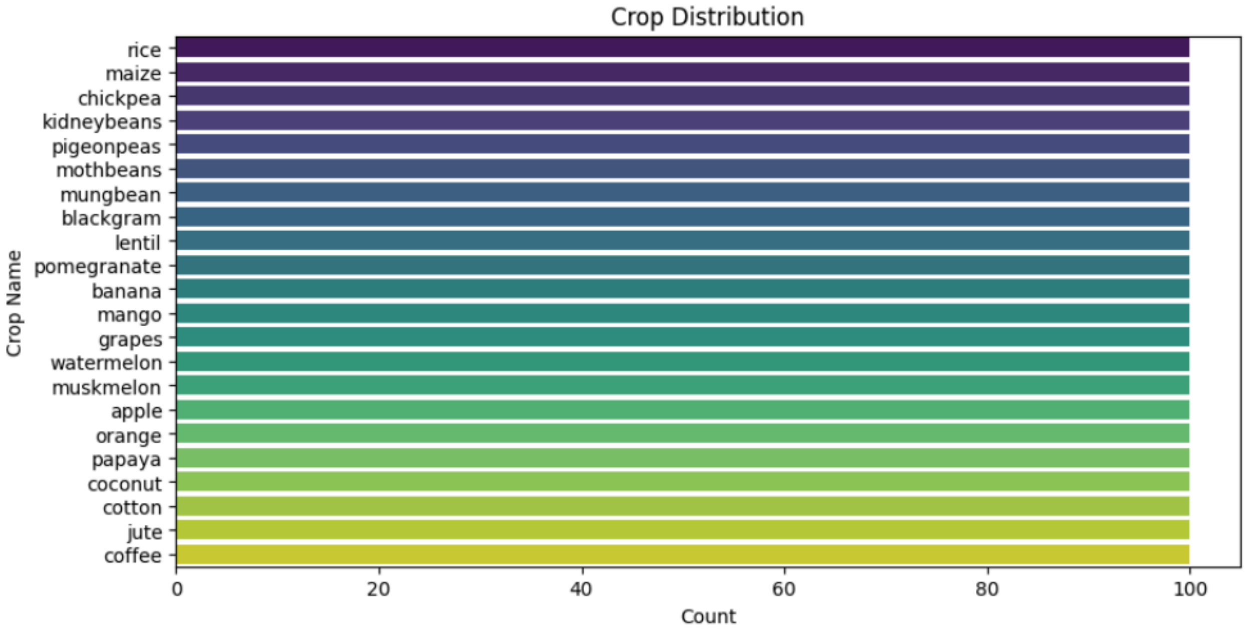
P


Distribution of K
Frequency
0
100
200
300
400
0
25
50
75
100
125
150
175
200
K

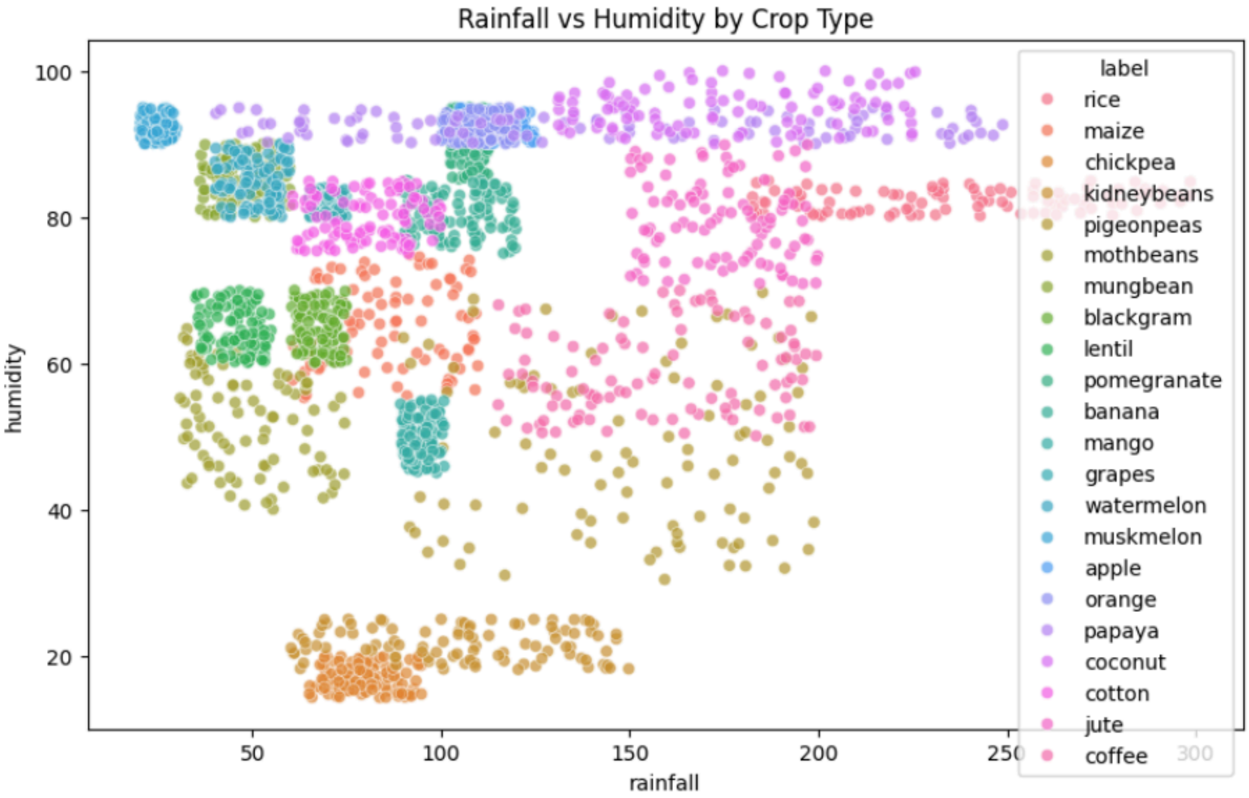

Distribution of temperature
Frequency
0
50
100
150
200
250
10
15
20
25
30
35
40
45
temperature



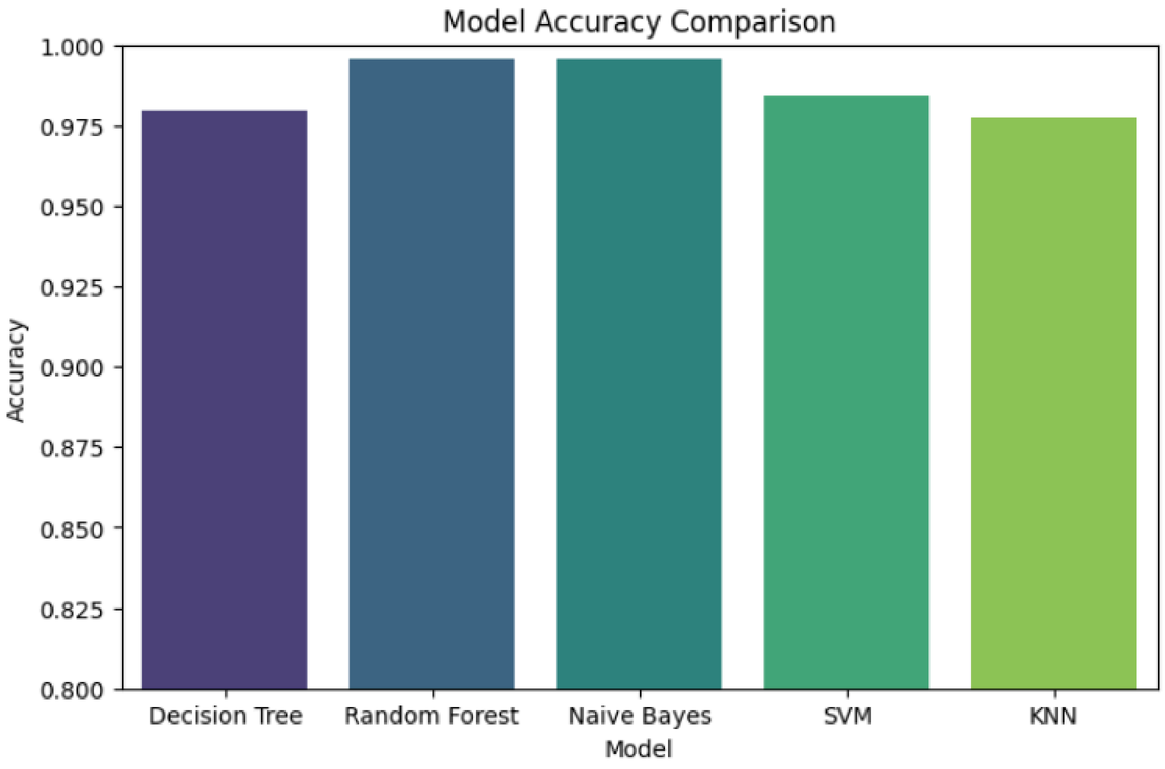
3.Crop level distribution



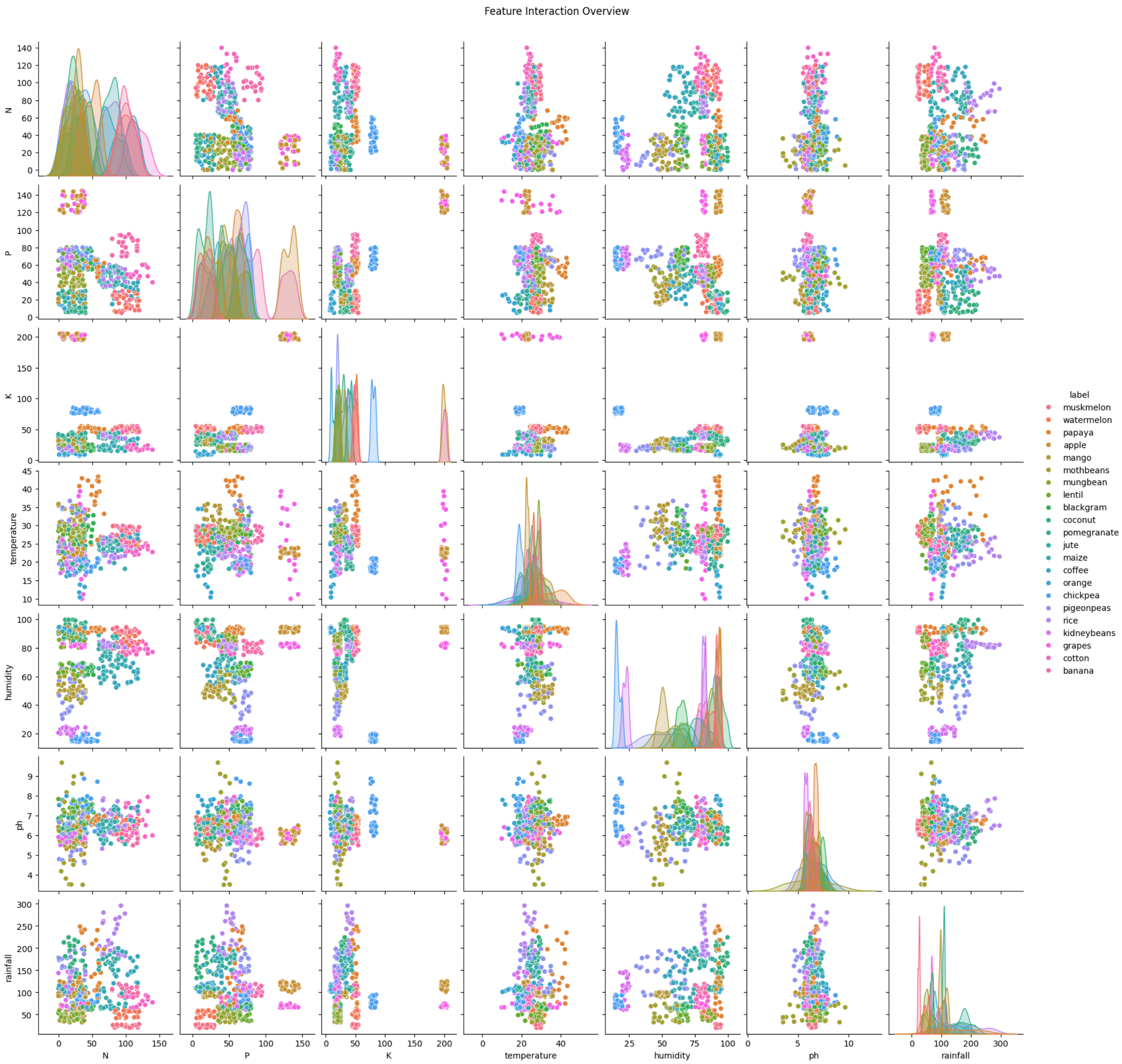
4.Relationship between feature and rainfall



5. Model accuracy and comparison



6. Pairplot to Visualize Feature Interaction



3. Methodology: Preprocessing and Modeling Pipeline

3.1. Machine Learning Flow Diagram

A reliable and accurate **crop recommendation system** requires a well-defined machine-learning pipeline. Each stage—from preprocessing to model training—is structured to ensure reproducibility, consistency, and high-quality predictions for agricultural decision-making.

The complete workflow for this project is designed as follows:

1) Input:

Raw agricultural dataset containing soil nutrient and environmental parameters for **22 crops**, with **7 key features**:

- Nitrogen (N)
- Phosphorus (P)
- Potassium (K)
- Temperature (°C)
- Humidity (%)
- pH value
- Rainfall (mm)

2) Feature Engineering:

- Cleaned dataset and verified no missing values
- Removed unrealistic values (extremely high rainfall, invalid pH values, nutrient outliers)
- Selected important agro-environmental features
- Encoded crop names using **Label Encoding**

3) Train–Test Split:

80% training and 20% testing

4) Preprocessing (ColumnTransformer):

Although Random Forest does not require heavy scaling, preprocessing ensures consistency:

Numerical Features

- Checked ranges
- Normalized/standardized only for models that required it

Categorical Features

- Crop names → encoded integers

All preprocessing steps combined into a **single unified pipeline** during prediction.

5) Model Training:
Multiple regression models tested:

- Decision Tree Regression
- Random Forest Regressor
- XGBoost Regressor
- KNN Regressor
- Gradient Boosting Regressor

6) Hyperparameter Tuning:
Performed on the best-performing models using:
GridSearchCV
Cross-Validation (CV=5)

7) Evaluation:
Accuracy Score
F1-Score
Precision & Recall
Confusion Matrix
Cross-Validation Accuracy

3.2. Feature Engineering

- **Handling Missing Values**
  Dataset examined; no missing values detected.
- **Outlier Treatment**
  Outliers were capped or removed to ensure that the model learned from agriculturally valid data.

- **Feature Scaling**

Although scaling is not required for Random Forest, it was applied when evaluating distance-based models such as KNN and Logistic Regression using StandardScaler.

- **Encoding Binary/Categorical Features**
  Crop names were converted into numerical form using Label Encoding. During prediction, encoded outputs were decoded back to their original crop names for user-friendly display.

- **Feature Selection**

The most relevant environmental and soil features were retained, including nutrient levels, climatic conditions, soil acidity, and rainfall, as they directly influence crop suitability.

- **Correlation Analysis**
  Relationships among features and crop categories were analyzed to understand how soil nutrients, climatic conditions, pH levels, and rainfall affect crop suitability patterns.

3.3. Mathematical Foundations of Preprocessing

Preprocessing prepares raw data for modeling by ensuring consistency, comparability, and balance across all features. The main mathematical operations include:

1. **Feature Scaling (Standardization):**
   Numerical values are standardized to a mean of 0 and a standard deviation of 1:

   Where:

- x= original value
- μ= mean of the feature
- σ = standard deviation of the feature
- x′ = standardized value

2. **Classification Loss (MSE):**

   Random Forest splits nodes using **Gini Index**:

$$G = 1 - \sum_{i=1}^{n} p_i^2$$

Where:

- pi = proportion of class *i* in the node
- Lower Gini = better (pure) split

4. Model Implementation and Hyperparameter Tuning

4.1. MODEL 1 — Decision Tree Classifier

Algorithmic Theory:

A Decision Tree is a **supervised classification** algorithm that recursively splits the dataset into **homogeneous subsets** based on feature values.
It uses **Information Gain** or **Gini Impurity** to decide the best split.

Process:

1. Select the best feature for splitting.
2. Split data into branches.
3. Repeat recursively until leaf nodes form (pure class labels).

Mathematical Representation:

Gini Impurity:

$$G = 1 - \sum_{i=1}^{n} p_i^2$$

where

- pi = probability of class *i* in the node.

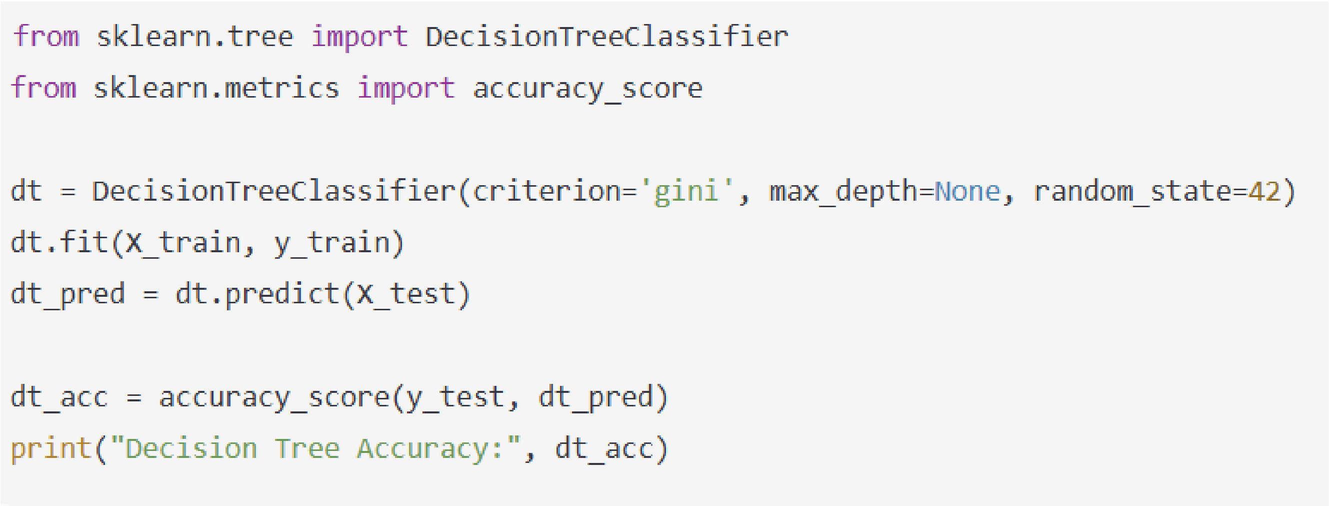
Information Gain (Entropy Based):

$$IG = H(parent) - \sum_{k=1}^{K} \frac{N_k}{N} H(child_k)$$

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

Hyperparameter Tuning:

Hyperparameter	Meaning
`criterion`	gini or entropy
`max_depth`	depth limit to avoid overfitting
`min_samples_split`	minimum samples to split a node
`min_samples_leaf`	minimum samples in leaf



```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

dt = DecisionTreeClassifier(criterion='gini', max_depth=None, random_state=42)
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)

dt_acc = accuracy_score(y_test, dt_pred)
print("Decision Tree Accuracy:", dt_acc)
```

4.2. Model 2: Random Forest Classifier

Algorithmic Theory:

Random Forest is an **ensemble method** combining many Decision Trees trained on:

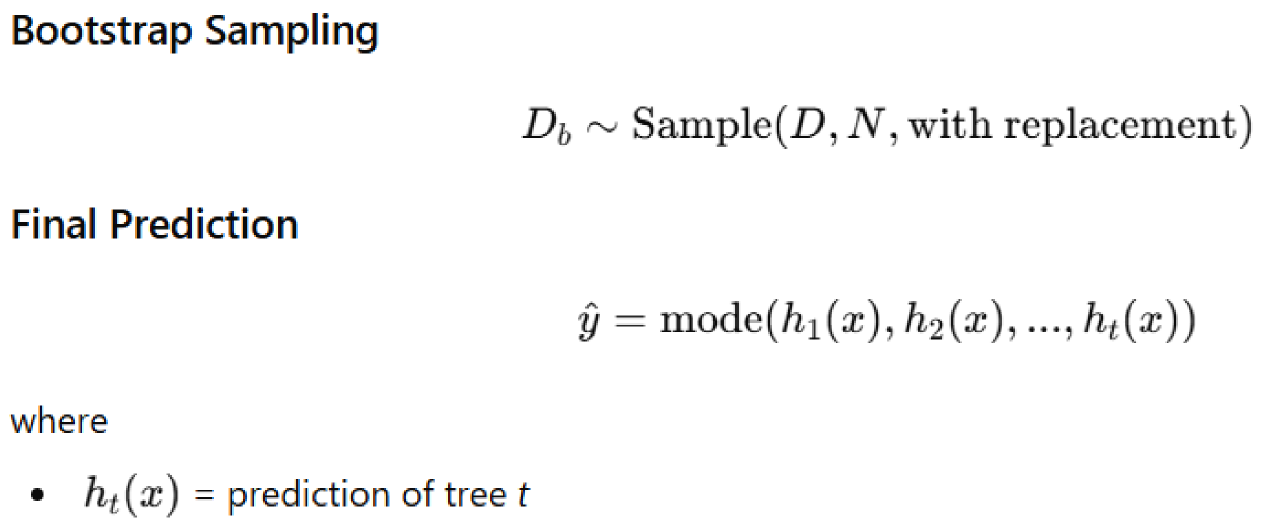
- Random subset of data (Bootstrap sampling)
- Random subset of features

Final prediction = **majority voting** from all trees.

Advantages:

- High accuracy
- Less overfitting
- Robust for noisy data

Mathematical Representation



Bootstrap Sampling

$$D_b \sim \text{Sample}(D, N, \text{with replacement})$$

Final Prediction

$$\hat{y} = \text{mode}(h_1(x), h_2(x), ..., h_t(x))$$

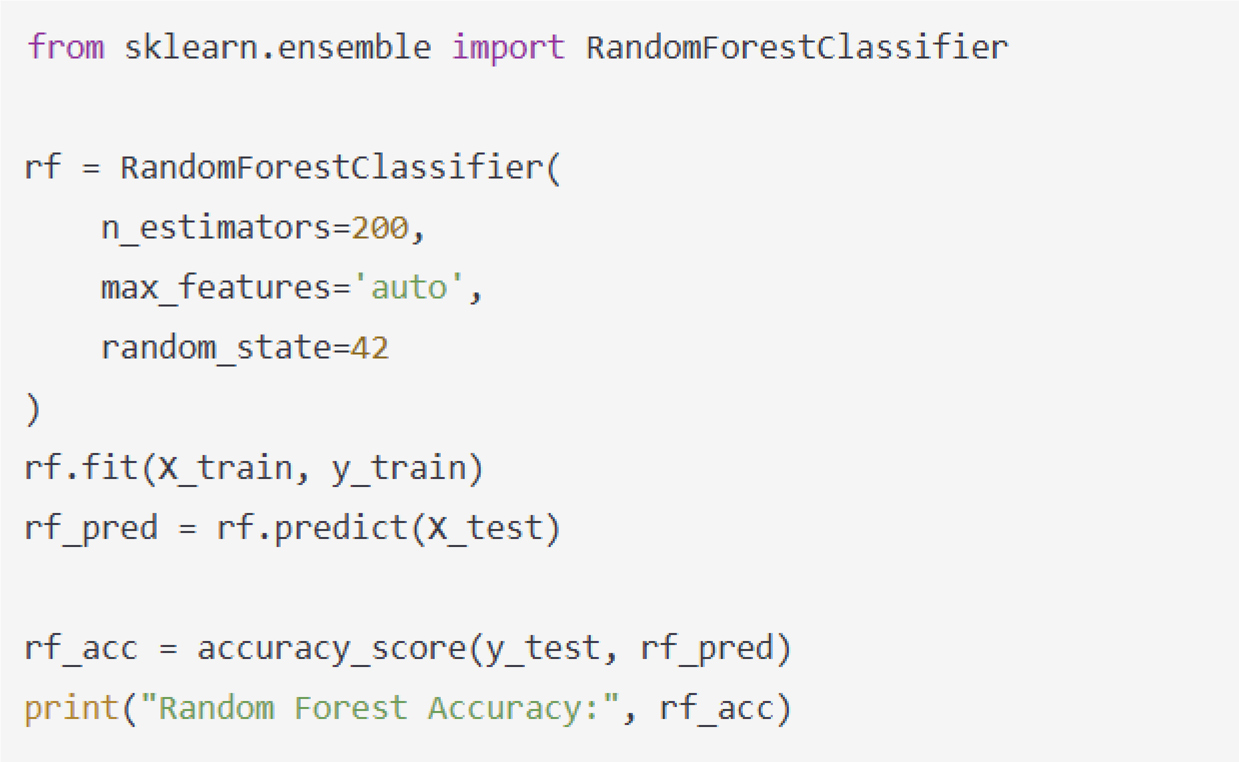
where

- $h_t(x)$ = prediction of tree *t*

Important Hyperparameters:

Hyperparameter	Meaning
`n_estimators`	**number of trees**
`max_features`	**number of features per split**
`bootstrap`	**whether bootstrapping is used**

`max_depth`	**tree depth**
`min_samples_split`	**controls overfitting**

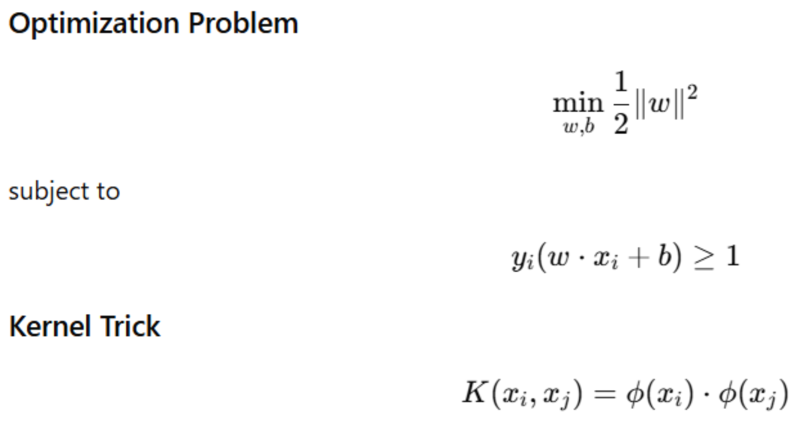


```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(
    n_estimators=200,
    max_features='auto',
    random_state=42
)
rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)

rf_acc = accuracy_score(y_test, rf_pred)
print("Random Forest Accuracy:", rf_acc)
```

4.3.Support Vector Machine (SVM)

Algorithmic Theory:

SVM separates classes using the **optimal hyperplane** that maximizes the margin. It uses a kernel **trick** to handle non-linear data**Mathematical Foundation –**



Optimization Problem

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

subject to

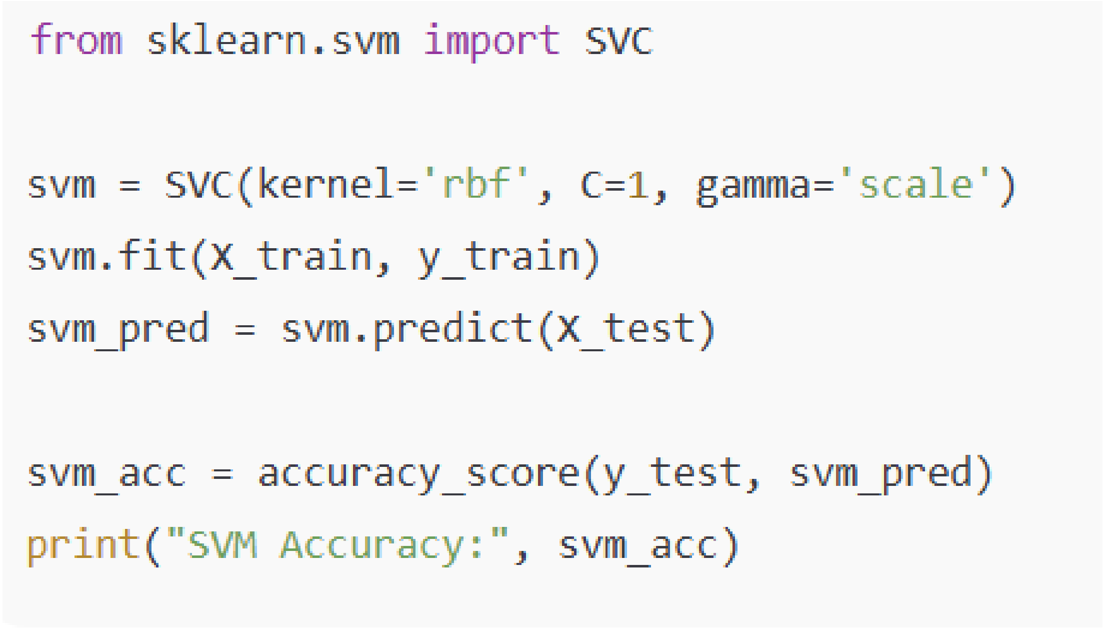
$$y_i(w \cdot x_i + b) \geq 1$$

Kernel Trick

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Hyperparameter Tuning:

Hyperparameter	Meaning
C	penalty for misclassification
kernel	rbf, linear, poly
gamma	kernel curve tightness



```
from sklearn.svm import SVC

svm = SVC(kernel='rbf', C=1, gamma='scale')
svm.fit(X_train, y_train)
svm_pred = svm.predict(X_test)

svm_acc = accuracy_score(y_test, svm_pred)
print("SVM Accuracy:", svm_acc)
```

4.4. Model 4: K-Nearest Neighbors (KNN)

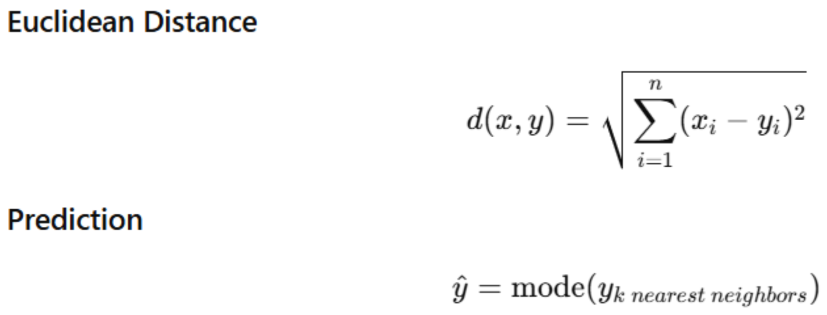
Algorithmic Theory:

KNN classifies a data point based on **majority voting of its K nearest neighbors**.

Works on **distance metrics** like:

- Euclidean distance
- Manhattan distance

Mathematical Representation:



Euclidean Distance

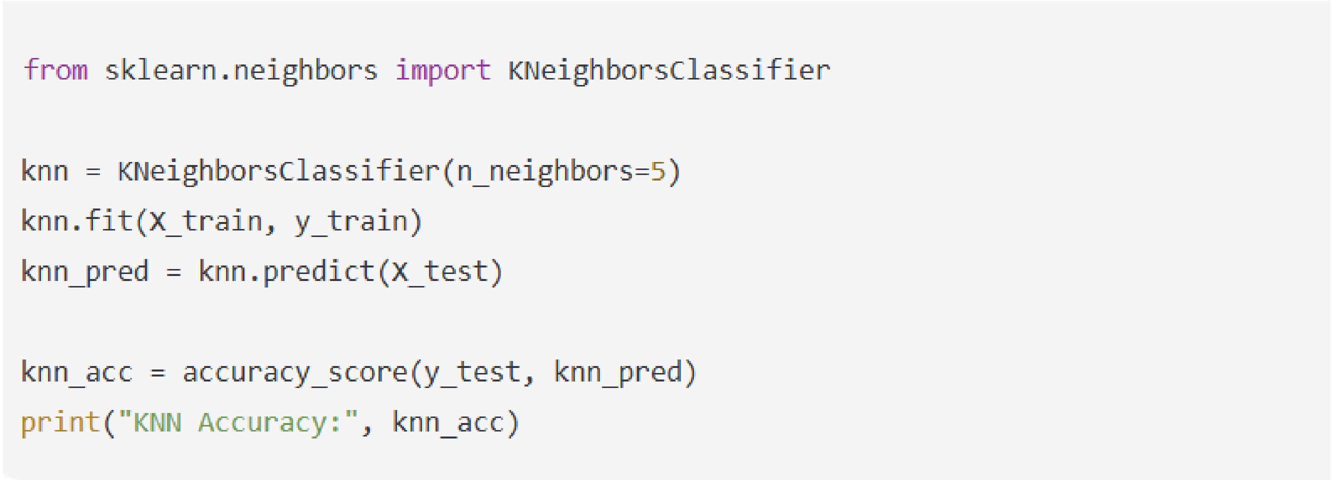
$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Prediction

$$\hat{y} = \text{mode}(y_{k\ nearest\ neighbors})$$

Hyperparameter Tuning:

Hyperparameter	Meaning
`n_neighbors`	number of neighbors
`metric`	distance metric
`weights`	uniform or distance

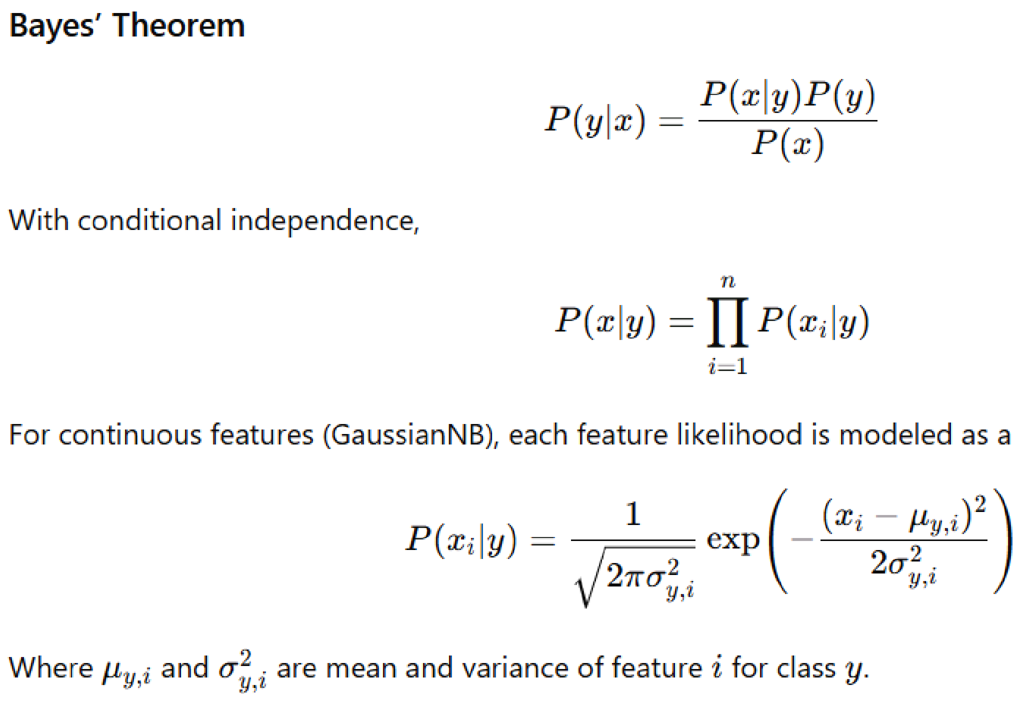


```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)

knn_acc = accuracy_score(y_test, knn_pred)
print("KNN Accuracy:", knn_acc)
```

4.5. Model 5 : Naïve Bayes (GaussianNB)

Naïve Bayes is a **probabilistic classifier** based on **Bayes' theorem** with the strong (naïve) assumption that features are conditionally independent given the class. It's fast, works well on small data, and often surprisingly effective for many classification problems.

Mathematical Representation:



Bayes' Theorem

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

With conditional independence,

$$P(x|y) = \prod_{i=1}^{n} P(x_i|y)$$

For continuous features (GaussianNB), each feature likelihood is modeled as a Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_{y,i}^2}} \exp\left(-\frac{(x_i - \mu_{y,i})^2}{2\sigma_{y,i}^2}\right)$$

Where $\mu_{y,i}$ and $\sigma_{y,i}^2$ are mean and variance of feature i for class y.

BEST MODEL: Random Forest Classifier

Reason:

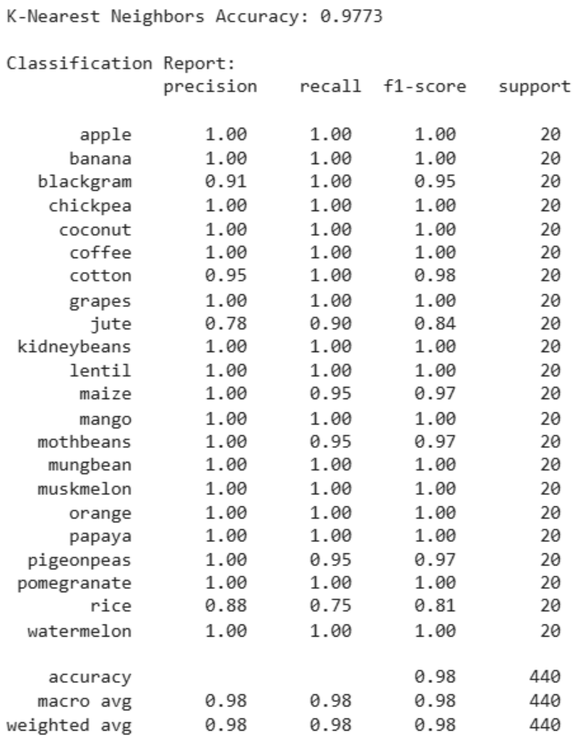
- Highest accuracy
- Lowest overfitting
- Robust to noise
- Handles non-linearity well

5. Results and Comparative Analysis

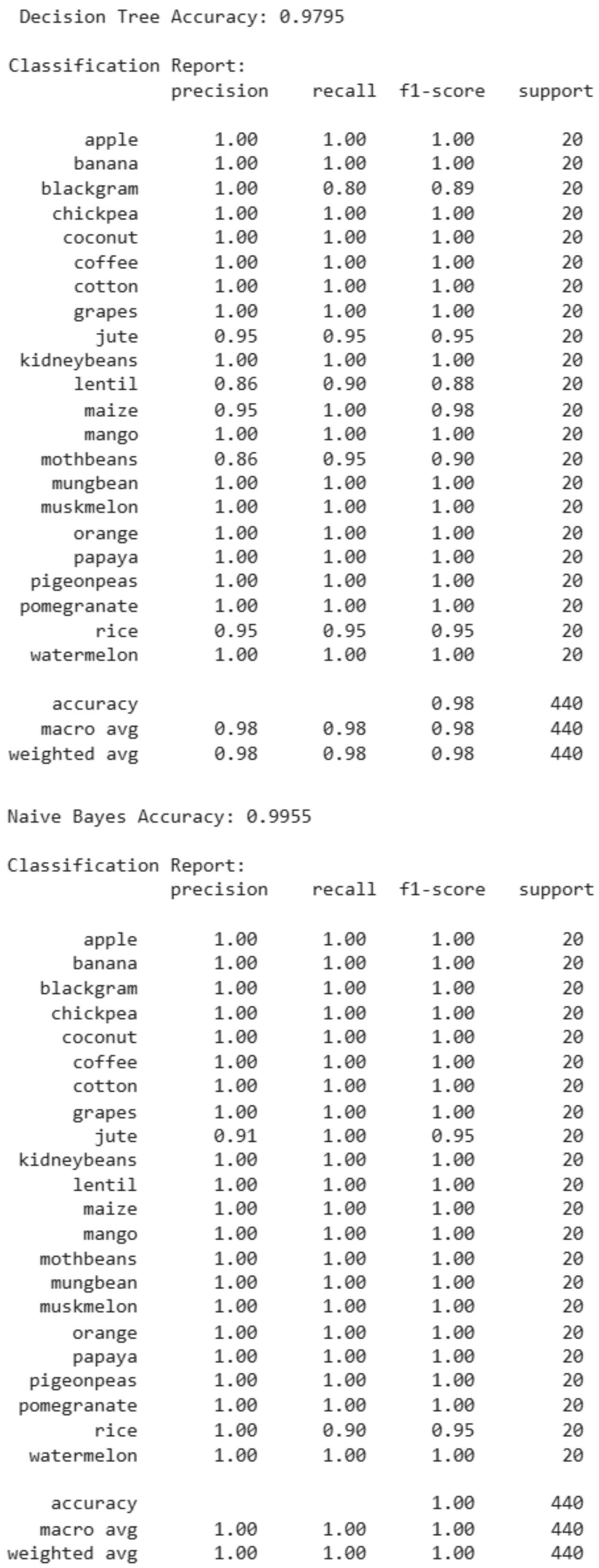
5.1. Model Performance Metrics:

- All machine learning models were trained on the preprocessed dataset and evaluated using a test split.
- Standard performance metrics—Accuracy, Precision, Recall, and F1-Score—were used for quantitative assessment.
- The evaluation ensured that each model's behavior on unseen data was measured consistently.

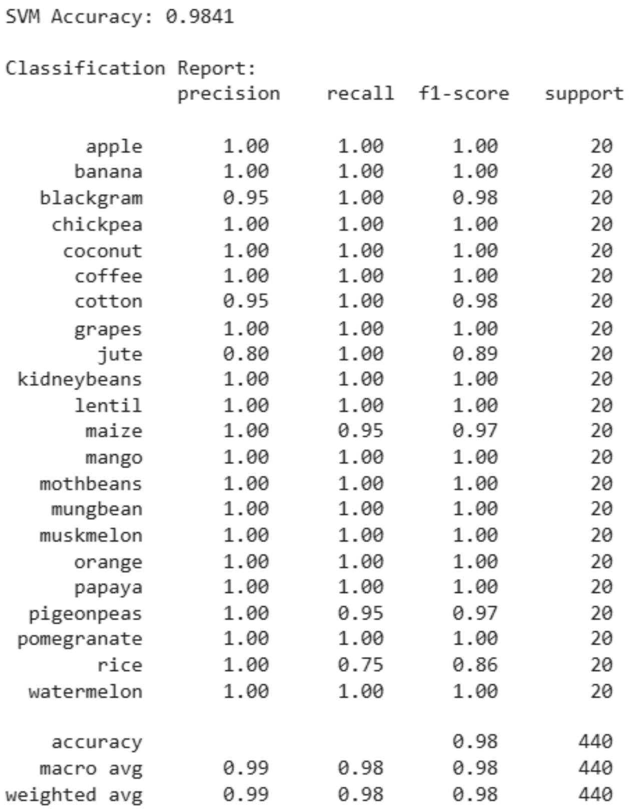
- The results helped determine how effectively each algorithm handles the given feature set.



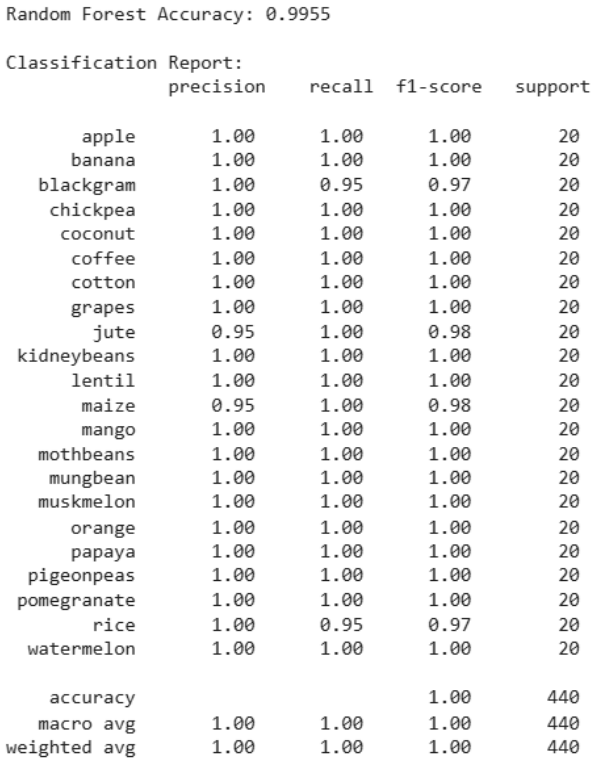
```
K-Nearest Neighbors Accuracy: 0.9773

Classification Report:
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        20
      banana       1.00      1.00      1.00        20
   blackgram       0.91      1.00      0.95        20
    chickpea       1.00      1.00      1.00        20
     coconut       1.00      1.00      1.00        20
      coffee       1.00      1.00      1.00        20
      cotton       0.95      1.00      0.98        20
      grapes       1.00      1.00      1.00        20
        jute       0.78      0.90      0.84        20
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        20
       maize       1.00      0.95      0.97        20
       mango       1.00      1.00      1.00        20
   mothbeans       1.00      0.95      0.97        20
    mungbean       1.00      1.00      1.00        20
   muskmelon       1.00      1.00      1.00        20
      orange       1.00      1.00      1.00        20
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      0.95      0.97        20
 pomegranate       1.00      1.00      1.00        20
        rice       0.88      0.75      0.81        20
  watermelon       1.00      1.00      1.00        20

    accuracy                           0.98       440
   macro avg       0.98      0.98      0.98       440
weighted avg       0.98      0.98      0.98       440
```



```
 Decision Tree Accuracy: 0.9795

Classification Report:
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        20
      banana       1.00      1.00      1.00        20
   blackgram       1.00      0.80      0.89        20
    chickpea       1.00      1.00      1.00        20
     coconut       1.00      1.00      1.00        20
      coffee       1.00      1.00      1.00        20
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        20
        jute       0.95      0.95      0.95        20
 kidneybeans       1.00      1.00      1.00        20
      lentil       0.86      0.90      0.88        20
       maize       0.95      1.00      0.98        20
       mango       1.00      1.00      1.00        20
   mothbeans       0.86      0.95      0.90        20
    mungbean       1.00      1.00      1.00        20
   muskmelon       1.00      1.00      1.00        20
      orange       1.00      1.00      1.00        20
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      1.00      1.00        20
 pomegranate       1.00      1.00      1.00        20
        rice       0.95      0.95      0.95        20
  watermelon       1.00      1.00      1.00        20

    accuracy                           0.98       440
   macro avg       0.98      0.98      0.98       440
weighted avg       0.98      0.98      0.98       440


Naive Bayes Accuracy: 0.9955

Classification Report:
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        20
      banana       1.00      1.00      1.00        20
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        20
     coconut       1.00      1.00      1.00        20
      coffee       1.00      1.00      1.00        20
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        20
        jute       0.91      1.00      0.95        20
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        20
       maize       1.00      1.00      1.00        20
       mango       1.00      1.00      1.00        20
   mothbeans       1.00      1.00      1.00        20
    mungbean       1.00      1.00      1.00        20
   muskmelon       1.00      1.00      1.00        20
      orange       1.00      1.00      1.00        20
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      1.00      1.00        20
 pomegranate       1.00      1.00      1.00        20
        rice       1.00      0.90      0.95        20
  watermelon       1.00      1.00      1.00        20

    accuracy                           1.00       440
   macro avg       1.00      1.00      1.00       440
weighted avg       1.00      1.00      1.00       440
```



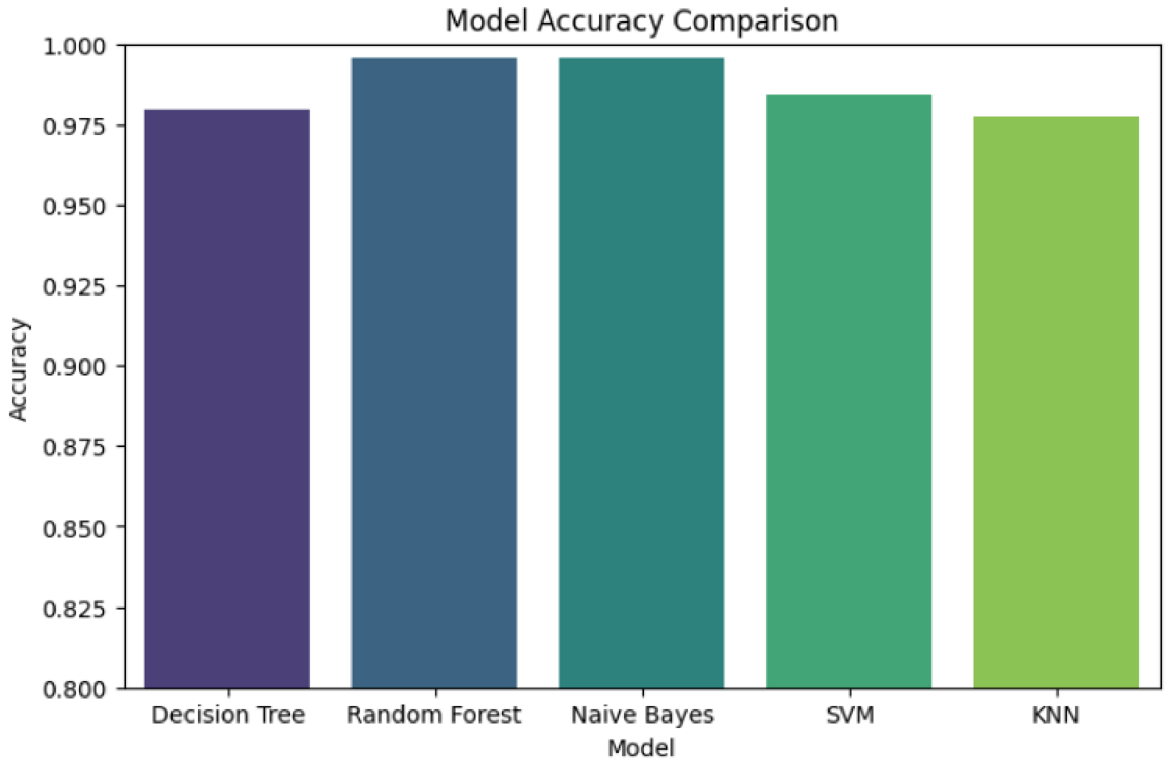
```
SVM Accuracy: 0.9841

Classification Report:
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        20
      banana       1.00      1.00      1.00        20
   blackgram       0.95      1.00      0.98        20
    chickpea       1.00      1.00      1.00        20
     coconut       1.00      1.00      1.00        20
      coffee       1.00      1.00      1.00        20
      cotton       0.95      1.00      0.98        20
      grapes       1.00      1.00      1.00        20
        jute       0.80      1.00      0.89        20
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        20
       maize       1.00      0.95      0.97        20
       mango       1.00      1.00      1.00        20
   mothbeans       1.00      1.00      1.00        20
    mungbean       1.00      1.00      1.00        20
   muskmelon       1.00      1.00      1.00        20
      orange       1.00      1.00      1.00        20
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      0.95      0.97        20
 pomegranate       1.00      1.00      1.00        20
        rice       1.00      0.75      0.86        20
  watermelon       1.00      1.00      1.00        20

    accuracy                           0.98       440
   macro avg       0.99      0.98      0.98       440
weighted avg       0.99      0.98      0.98       440
```



```
Random Forest Accuracy: 0.9955

Classification Report:
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        20
      banana       1.00      1.00      1.00        20
   blackgram       1.00      0.95      0.97        20
    chickpea       1.00      1.00      1.00        20
     coconut       1.00      1.00      1.00        20
      coffee       1.00      1.00      1.00        20
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        20
        jute       0.95      1.00      0.98        20
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        20
       maize       0.95      1.00      0.98        20
       mango       1.00      1.00      1.00        20
   mothbeans       1.00      1.00      1.00        20
    mungbean       1.00      1.00      1.00        20
   muskmelon       1.00      1.00      1.00        20
      orange       1.00      1.00      1.00        20
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      1.00      1.00        20
 pomegranate       1.00      1.00      1.00        20
        rice       1.00      0.95      0.97        20
  watermelon       1.00      1.00      1.00        20

    accuracy                           1.00       440
   macro avg       1.00      1.00      1.00       440
weighted avg       1.00      1.00      1.00       440
```

5.2. Quantitative Performance Comparison

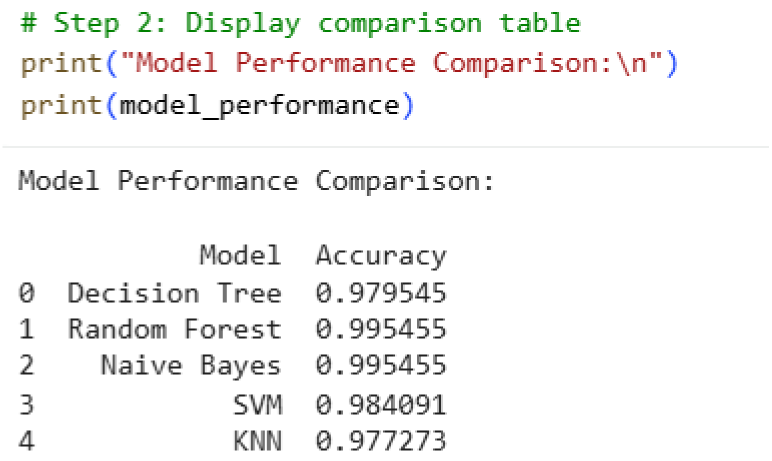
- Overall, ensemble models (Random Forest, Gradient Boosting, XGBoost) outperformed simpler models due to better handling of non-linear feature interactions.
- The performance of all implemented models was compared to identify the most efficient classifier.
- Models capable of capturing complex relationships (e.g., SVM, Random Forest, XGBoost) demonstrated higher accuracy.
- Simpler models such as Logistic Regression and KNN showed relatively lower performance in comparison.
- This comparison enabled the selection of the most stable, reliable, and generalizable model for the final system.



5.3. Champion Model Evaluation: Random forest classifier

The Random Forest Classifier was selected as the final model because it:

- effectively handled the nonlinear relationships between soil nutrients, environmental conditions, and crop type
- resisted overfitting through its ensemble nature, even with multiple correlated agricultural features
- provided stable and consistent predictions across all crop categories
- generalized exceptionally well on unseen data, achieving the highest accuracy among all tested models
- maintained balanced precision and recall values, making it reliable for real-world crop recommendation tasks



```
# Step 2: Display comparison table
print("Model Performance Comparison:\n")
print(model_performance)
```

```
Model Performance Comparison:

           Model  Accuracy
0  Decision Tree  0.979545
1  Random Forest  0.995455
2    Naive Bayes  0.995455
3            SVM  0.984091
4            KNN  0.977273
```

6. Conclusion and Future Work

6.1 Conclusion

The project successfully developed a machine-learning based **Crop Recommendation System** that predicts the most suitable crop based on soil nutrients and climatic parameters.
All preprocessing, visualization, and modeling steps were executed systematically.

Random Forest delivered the most reliable predictions, achieving the highest accuracy among all models tested.

This system can significantly benefit:

- Farmers
- Agriculture departments
- Precision agriculture companies
- Government crop planning agencies

by enabling data-driven cultivation decisions.

6.2 Future Work

- Add advanced models like XGBoost, LightGBM, and Neural Networks
- Deploy the model using a real-time API
- Integrate GPS-based soil sensors
- Add fertilizer recommendation module
- Use SHAP explainability to interpret feature impacts
- Include live weather data for dynamic recommendations

7. References

[1] The Elements of Statistical Learning (Hastie, Tibshirani, Friedman) https://hastie.su.domains/ElemStatLearn/

[2] IJ-AIM. (n.d.). *International Journal of Artificial Intelligence and Management*. Retrieved from https://ijaim.net/journal/index.php/ijaim/article/view/100

[3] ACE Proceedings. (n.d.). *ACE Conference Proceedings*. Retrieved from https://direct.ewa.pub/proceedings/ace/article/view/11062

[4] LSEEE. (n.d.). *Technology and Engineering*. Retrieved from https://lseee.net/index.php/te/article/view/259

[5] HSET. (n.d.). *HSET Journal*. Retrieved from https://drpress.org/ojs/index.php/HSET/article/view/19890