**CROP PREDICTION USING MACHINE LEARNING**

* **Objectives**

The primary objective of this project is to build an intelligent system that

recommends the most suitable crop based on environmental and soil

parameters using machine learning algorithms. The system leverages Random

Forest and XGBoost classifiers to accurately predict the ideal crop, thereby

supporting farmers and agricultural stakeholders in decision-making.

The main objectives of this project are:

* To develop a machine learning model that can predict the most suitable crop for cultivation based on input features such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall.
* To compare the performance of two powerful classification algorithms: Random Forest and XGBoost.
* To provide a user-friendly interface for users to input data and receive crop recommendations.
* To export and deploy the model using ONNX for cross-platform support, especially for mobile/embedded applications.
* **Introduction**

With the advancement of artificial intelligence and data science, machine learning

models can be used in the agriculture sector to enhance productivity and decision

making. Crop prediction is one such area where the correct choice of crop based on

environmental and soil conditions can lead to better yield and profitability. In this

project, we developed a crop recommendation system using two machine learning

algorithms: Random Forest and XGBoost. The system processes user input through

a trained model and returns the best-suited crop for cultivation.

* **Technology Used**
* **Programming Language: Python**
* **Libraries:** Pandas, NumPy, Scikit-learn, XGBoost, Joblib, ONNX
* **IDE:** Jupyter Notebook
* **Data Visualization:** Confusion matrix, classification report
* **Model Deployment:** ONNX runtime
* **Design and Implementation**

1. **Data Loading and Exploration-**

The dataset used contains 2200 entries with 23 columns, including features such as N, P, K, temperature, humidity, pH, and rainfall. The target variable is the 'Crop'.

**Code**- import pandas as pd

import numpy as np

df = pd.read\_csv("Crop\_recommendationV2.csv")

print(df.info())

print(df.head())

1. **Feature Selection**-

Selected features for prediction:

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

**Code-** X = df[['N','P','K','temperature','humidity','ph','rainfall']]

y = df['Crop']

1. **Data Preprocessing-**
   * Loaded using pandas.read\_csv().
   * Label encoding was applied to the target variable Crop using LabelEncoder.
   * The dataset was split into train and test sets using train\_test\_split().
   * Standardization was applied using StandardScaler.

**Code-** from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded,

test\_size=0.2, random\_state=42, stratify=y)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

1. **Label Encoding**-

Crops are categorical values; hence, LabelEncoder is used to convert them to

numeric form.

**Code-** from sklearn.preprocessing import LabelEncoder

label\_encoder = LabelEncoder()

y\_encoded = label\_encoder.fit\_transform(y)

1. **Model Training-**
   * **Random Forest Classifier**: Trained with 100 estimators, random state = 42.

**Code-** from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n\_estimators=100,

random\_state=42)

model.fit(X\_train, y\_train)

* + **XGBoost Classifier**: Trained with 100 estimators, learning rate = 0.1, max depth = 6.

**Code-** from xgboost import XGBClassifier

model = XGBClassifier(n\_estimators=100, learning\_rate=0.1,

max\_depth=6, random\_state=42, use\_label\_encoder=False,

eval\_metric='mlogloss')

model.fit(X\_train, y\_train)

1. **Evaluation-**
   * Model performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrix.
   * Accuracy obtained: **98.86%**

**Code-** from sklearn.metrics import accuracy\_score, classification\_report,

confusion\_matrix

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

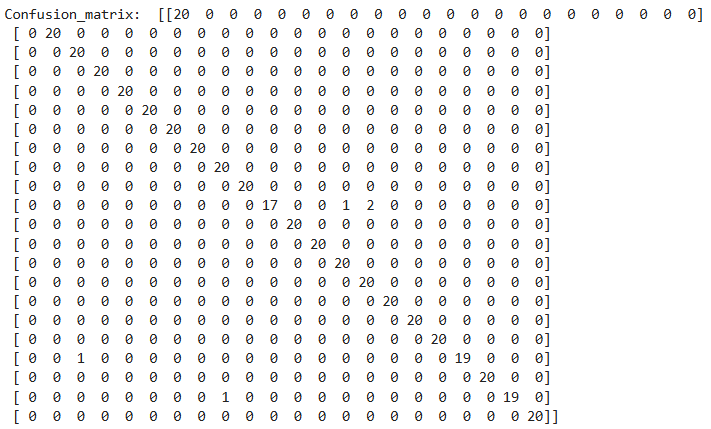
print(classification\_report(y\_test, y\_pred))

print("Confusion\_matrix:", confusion\_matrix(y\_test, y\_pred))

1. **Sample Output-**

* Accuracy: 0.9886 (XGBoost)
* Classification Report:
  + Precision, recall, and F1-scores are mostly above 0.95 across classes.
  + Support per class: 20 samples

1. **Confusion Matrix-**



1. **Classification Report Sample-**
   * Accuracy: 0.9886
   * Precision: 0.99
   * Recall: 0.99
   * F1-score: 0.99
   * Support: 440
2. **Prediction Function-**
   * A function predict\_crop() was created to take real-time user input.
   * The input is scaled and passed to the trained model.
   * Example Input:
   * Nitrogen (N): 50
   * Phosphorus (P): 40

**Code-** def predict\_crop():

input\_features = []

feature\_names = ["Nitrogen (N)", "Phosphorus (P)", "Potassium (K)",

"Temperature (C)", "Humidity (%)", "pH Level",

"Rainfall (mm)"]

for feature in feature\_names:

value = float(input(f"Enter {feature}: "))

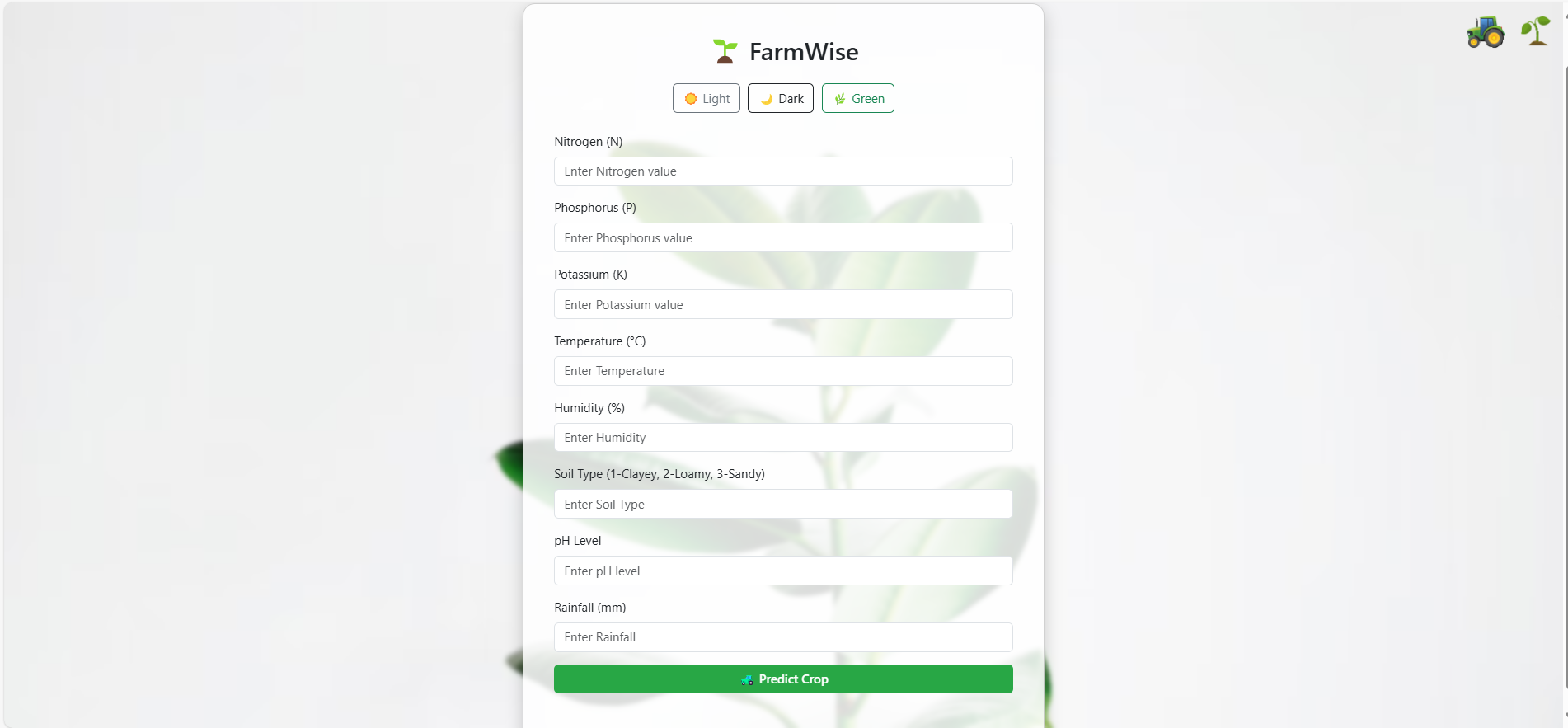
input\_features.append(value)

input\_scaled = scaler.transform([input\_features])

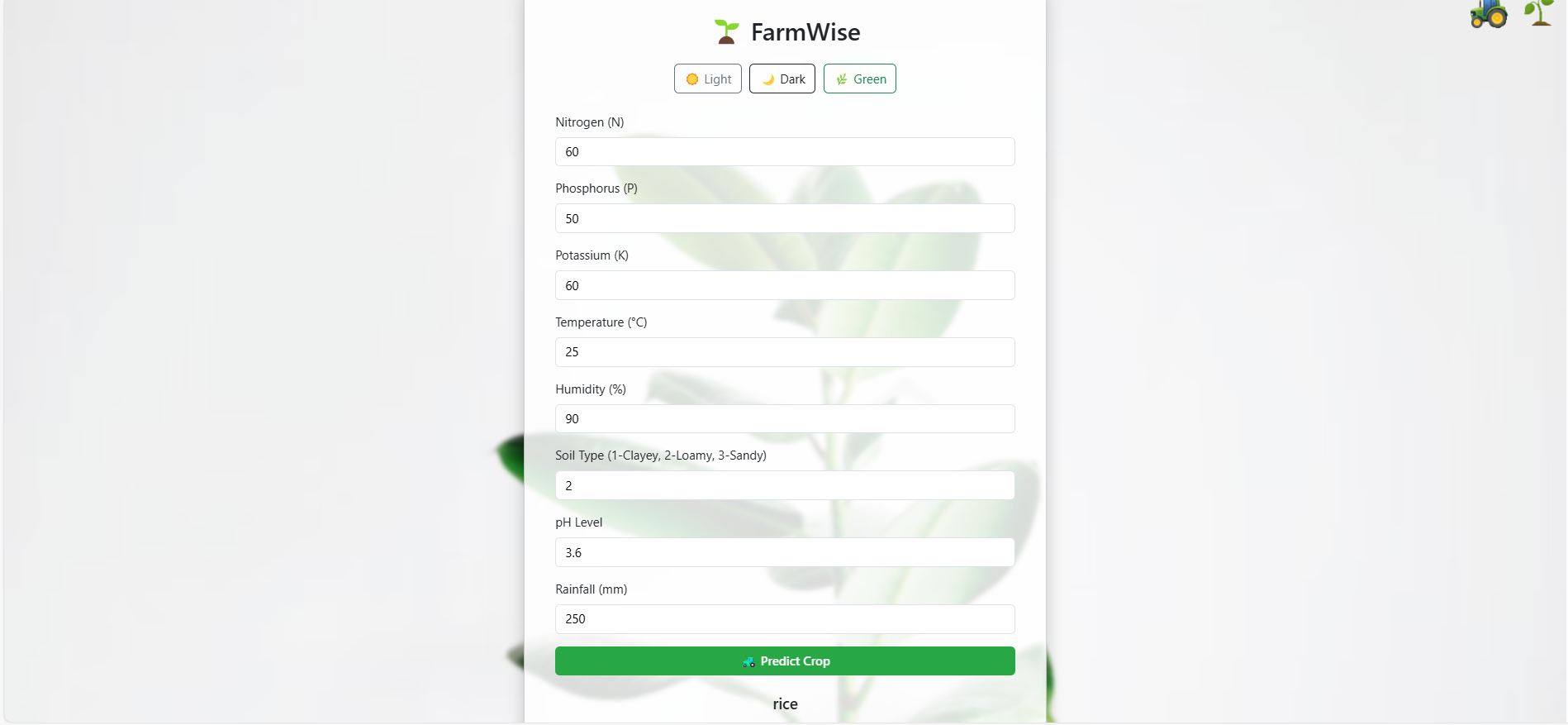
predicted\_crop = model.predict(input\_scaled)[0]

print(f"\nRecommended Crop: {predicted\_crop}")

1. **Sample Input and Output with UI-**



**Output**



1. **Model Export-**
   * The model and scaler were saved using joblib for future use.
   * Converted to ONNX format for deployment:
   * initial\_type = [("input", FloatTensorType([None, X.shape[1]]))]
   * onnx\_model = convert\_sklearn(model, initial\_types=initial\_type)
   * Model loaded and tested with onnxruntime.
   * Input and output names extracted for mobile app use.
2. **Label Mapping for App Development-**

Crop label index mapping:

* 0: apple
* 1: banana
* 2: blackgram
* 3: chickpea
* 4: coconut
* 5: coffee
* 6: cotton
* 7: grapes
* 8: jute
* 9: kidneybeans
* 10: lentil
* 11: maize
* 12: mango
* 13: mothbeans
* 14: mungbean
* 15: muskmelon
* 16: orange
* 17: papaya
* 18: pigeonpeas
* 19: pomegranate
* 20: rice
* 21: watermelon
* **Conclusion** -

The crop prediction system using Random Forest and XGBoost models

achieved a very high accuracy of nearly 99%. The system is capable of

assisting farmers and agricultural professionals in making informed decisions

regarding crop selection. With features like real-time prediction, model

export, and cross-platform compatibility through ONNX, this model has the

potential to be deployed in real-world smart agriculture applications and

embedded systems like Android or IoT platforms. Future work can focus on

expanding the feature set, integrating live weather and soil data, and

developing a full mobile or web-based interface for ease of use.