**Crop Prediction Web APP Code**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

import joblib

# Download crop dataset from Kaggle or use your own

!wget https://raw.githubusercontent.com/just4nitesh/Crop-Recommendation-System-ML/main/crop\_recommendation.csv

# Load dataset

data = pd.read\_csv("crop\_recommendation.csv")

X = data.drop("label", axis=1)

y = data["label"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train Random Forest model

model = RandomForestClassifier()

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

# Save the model

joblib.dump(model, "crop\_model.pkl")

**Create Streamlit App File**

python

%%writefile app.py

import streamlit as st

import numpy as np

import joblib

model = joblib.load("crop\_model.pkl")

st.title("🌾 Smart Crop Predictor")

st.markdown("Enter the soil and weather information to predict the best crop.")

N = st.number\_input("Nitrogen (N)", 0, 150)

P = st.number\_input("Phosphorous (P)", 0, 150)

K = st.number\_input("Potassium (K)", 0, 150)

temperature = st.number\_input("Temperature (°C)", 0.0, 50.0)

humidity = st.number\_input("Humidity (%)", 0.0, 100.0)

ph = st.number\_input("Soil pH", 0.0, 14.0)

rainfall = st.number\_input("Rainfall (mm)", 0.0, 300.0)

if st.button("Predict Crop"):

features = np.array([[N, P, K, temperature, humidity, ph, rainfall]])

prediction = model.predict(features)

st.success(f"🌱 Recommended Crop: \*\*{prediction[0].capitalize()}\*\*")