Crop prediction for rice and wheat involves using various data sources like weather conditions, soil quality, temperature, rainfall, and historical yield data to estimate future crop yields. Machine learning models, remote sensing, and GIS-based techniques are commonly used to improve accuracy.

Factors Affecting Crop Prediction

1. **Climate Factors**: Temperature, rainfall, humidity, wind speed, and solar radiation.
2. **Soil Factors**: pH levels, nutrient availability, moisture content, and organic matter.
3. **Agricultural Practices**: Sowing time, irrigation techniques, fertilizers, and pest control.
4. **Remote Sensing & Satellite Data**: NDVI (Normalized Difference Vegetation Index) and other vegetation indices help monitor crop health.
5. **Historical Yield Data**: Past yield records are used to forecast future production.

Technologies Used in Crop Prediction

1. **Machine Learning & AI Models**:
    - Random Forest, Decision Trees, SVM, and Deep Learning models (ANN, CNN, LSTMs) for predictive analysis.
    - Regression models to estimate yield based on weather and soil data.
2. **Satellite & Drone-Based Imaging**:
    - Uses hyperspectral and multispectral imaging to assess crop health.
3. **IoT & Smart Sensors**:
    - Real-time data collection from soil moisture sensors, weather stations, and irrigation systems.
4. **GIS & Remote Sensing**:
    - Spatial analysis of land use and crop mapping.

Challenges in Crop Prediction

- Unpredictable climate changes.
- Data availability and accuracy issues.
- Pest and disease outbreaks.
- Soil degradation and water scarcity.

To build a **machine learning model for crop prediction on rice and wheat**, we can follow these steps:

1. Collect & Prepare Data

We need datasets that include:

- **Climate data**: Temperature, rainfall, humidity, wind speed.
- **Soil data**: pH, nitrogen, phosphorus, potassium levels.
- **Agricultural practices**: Sowing date, irrigation, fertilizers.
- **Historical yield data**: Past production records.

Sources for datasets:

- **FAO (Food and Agriculture Organization)**
- **NASA Earth Data**
- **Indian Agricultural Data Portal**
- **Kaggle (Crop Yield Prediction datasets)**

---

2. Choose ML Model

Possible models:

1. **Regression Models** (for yield prediction)
    - Linear Regression
    - Random Forest Regressor
    - XGBoost
2. **Classification Models** (to classify high/low yield areas)
    - Decision Trees
    - Support Vector Machines (SVM)
    - Neural Networks (ANN)

---

3. Implement Machine Learning Model

We'll use Python with **Pandas, Scikit-learn, TensorFlow, and Matplotlib**.

Would you like:

1. **A simple regression model for yield prediction?**
2. **A classification model to predict high/low yield areas?**
3. **A deep learning model for advanced prediction?**

Here's a **machine learning model for crop yield prediction** using **Random Forest Regression**. It predicts the yield of **rice and wheat** based on climate, soil, and farming data.

Steps in the Code:

1. Load the dataset
2. Preprocess the data (handle missing values, encode categorical data)
3. Train a **Random Forest Regressor**
4. Evaluate the model and make predictions

Python Code for Crop Yield Prediction

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler
```

```
from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score


# Load the dataset (Replace 'crop_data.csv' with your dataset file)

df = pd.read_csv("crop_data.csv")


# Display the first few rows

print(df.head())


# Check for missing values

print(df.isnull().sum())


# Drop missing values (or handle them using imputation)

df.dropna(inplace=True)


# Selecting Features (X) and Target Variable (y)

features = ['temperature', 'rainfall', 'humidity', 'soil_pH', 'nitrogen', 'phosphorus', 'potassium']

target = ['yield']  # Crop yield column


X = df[features]

y = df[target]


# Split data into training and testing sets (80% train, 20% test)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a Random Forest Regression Model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train.values.ravel())

# Make Predictions
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
print(f"R2 Score: {r2}")

# Plot Actual vs Predicted values
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel("Actual Yield")
plt.ylabel("Predicted Yield")
```

```
plt.title("Actual vs Predicted Crop Yield")

plt.show()
```

How to Use the Code

- **Prepare your dataset** (crop_data.csv) with columns like:
  temperature, rainfall, humidity, soil_pH, nitrogen, phosphorus, potassium, yield
- **Run the script** in Python.
- It will train a **Random Forest model** and predict yield based on input features.
- **Performance metrics** (MAE, MSE, R^2) will be displayed.
- The **scatter plot** will compare actual vs predicted yields.

Next Steps

- **Improve accuracy** by tuning hyperparameters (n_estimators, max_depth, etc.).
- **Try deep learning** (ANN/LSTM) for more accurate predictions.
- **Deploy the model** using Flask or FastAPI.