Crop Recommendation System

Project Report

<Version 1.0>

Industrial Project (IDS851)

Degree

BACHELOR OF TECHNOLOGY (CSE)

PROJECT GUIDE:

Mr. Ajay Rastogi

Mr. Divyanshu Saxena

SUBMITTED BY:

Sahil Jain (TCA2166016)

June, 2025

FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

ACKNOWLEDGEMENT

I would like to express my heartfelt thanks to everyone who helped and supported me throughout the "Crop Recommendation System Using Machine Learning" project.

First, I want to thank my project supervisor for their continuous guidance and advice, which helped me stay on track. I'm also very grateful to the faculty members and technical team for their valuable feedback and support.

A big thank you to the data providers and agricultural experts whose data and knowledge made this project possible. I also appreciate the developers and open-source community for providing tools like Python, scikit-learn which made building the system so much easier.

Everyone's help has been crucial to completing this project, and I am deeply grateful for all the support I received.

Sahil Jain

TCA2166016

Place: Moradabad

Date: 05\06\2025

DECLARATION

We hereby declare that this Project Report titled **Crop Recommendation System** submitted by us and approved by our project guide, the College of Computing Sciences and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Project Group :	Project Group Name/Id	
Student Name:	Sahil Jain	Signature
Project Guide: (External)	Name	Signature
Project Guide: (Internal)	Mr. Ajay Rastogi Mr. Divyanshu Saxena	Signature

Brief About the Company

CodeApto is a software development company where we specialize in pioneering transformative processes tailored to meet the unique challenges faced by startups worldwide. As leaders in bespoke consulting and development, we are dedicated to delivering innovative solutions that address specific needs and complexities across various industries.
At CodeApto, we redefine possibilities through customized strategies and cutting-edge solutions designed to foster growth and sustainability. Our approach combines deep industry expertise with a keen understanding of local nuances, regulatory landscapes, and emerging trends in sectors such as technology, e-commerce, and beyond. We empower startups with adaptable, localized solutions that not only meet immediate challenges but also anticipate future needs. From navigating regulatory requirements to leveraging advanced technologies, we provide comprehensive support and guidance that aligns with the dynamic nature of business ecosystems globally.
Join us in shaping the future of startups through bespoke consulting that blends innovation, customization, and strategic insight. Together, we can overcome obstacles, seize opportunities, and drive meaningful impact in the rapidly evolving landscape of entrepreneurship.

Website
https://www.codeapto.com/

Industry
Technology, Information and Internet

Company size
11-50 employees

Headquarters
Bangalore, Karnataka

Table of Contents

Appendix

A: Data Flow Diagram (DFD)

B: Entity Relationship Diagram (ERD)

C: Use Case Diagram (UCD)

D: Data Dictionary (DD)

E: Screen Shots

1 Project Title

Crop Recommendation System Using Machine Learning - This project is designed to assist farmers in selecting the most suitable crops to grow based on environmental factors like soil nutrients, weather conditions, and historical data. By leveraging machine learning algorithms, the system provides tailored recommendations that improve crop yields, optimize resource usage, and contribute to sustainable agricultural practices.

2 Problem Statement

The agriculture industry faces challenges in determining the most suitable crops to grow in specific regions based on environmental data. The problem is compounded by changing climate conditions, which further complicate the decision-making process for farmers. This project aims to develop a machine learning-based crop recommendation system that helps farmers predict the best crops to grow based on environmental parameters.

3 Project Description

This Project is an intelligent system designed to help farmers make data-driven decisions regarding which crops to grow based on the environmental conditions of their specific region. By leveraging machine learning algorithms and historical environmental data, the system aims to provide accurate crop recommendations that improve productivity, optimize resource usage, and contribute to sustainable farming practices.

3.1 Scope of the Work

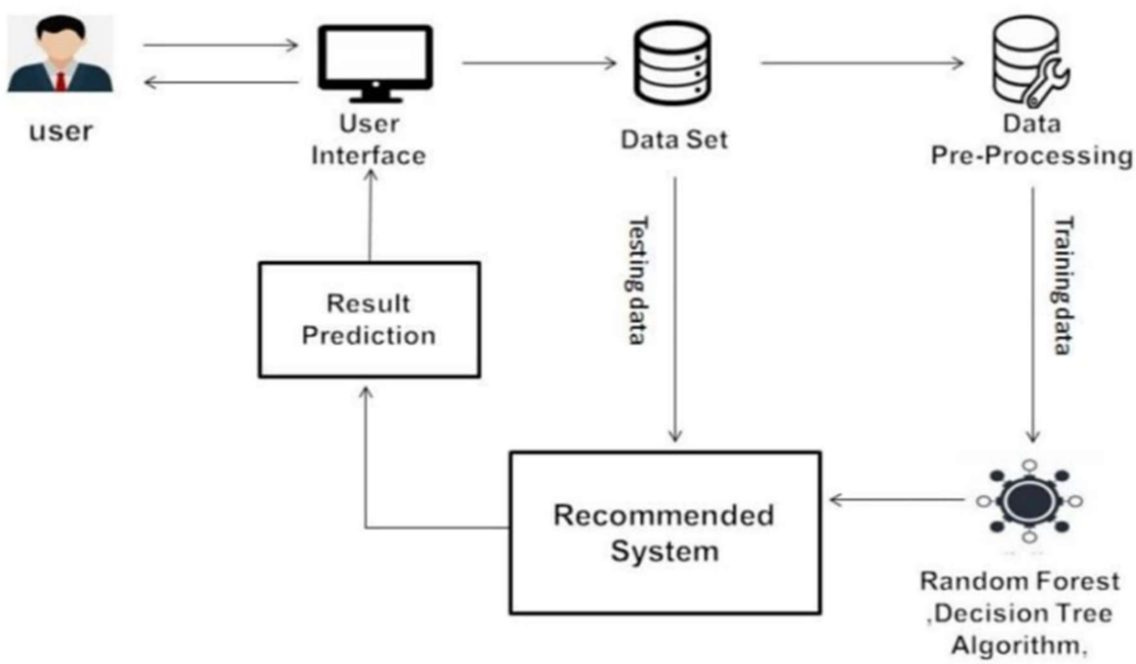
- **Data Collection and Preprocessing**: The system collects environmental data, including soil nutrient levels (N, P, K), temperature, humidity, pH, and rainfall. The data is cleaned and pre-processed to remove inconsistencies and ensure it is ready for use in model training.
- **Machine Learning Model Development**: Various machine learning models (including Random Forest, Decision Trees, Logistic Regression, Naive Bayes, etc.) will be trained on historical environmental data to predict the most suitable crops for a given environment. Model accuracy will be assessed, and the best-performing model will be selected for deployment.
- **Crop Prediction**: The final model will be used to predict the most suitable crop based on new environmental input data. The system will provide recommendations based on data entered by the user (e.g., a farmer entering values for temperature, humidity, etc.).

- **User Interface**: A user-friendly interface will allow farmers to input environmental data, view crop recommendations, and receive real-time predictions based on the most recent data. This interface may include options to save and track historical recommendations.
- **Admin Panel**: The admin panel will allow administrators to manage crop and environmental data, retrain models, update system parameters, and view system performance.

3.2 Project Modules

- **Data Collection**: Collect data related to environmental factors such as nitrogen (N), phosphorous (P), potassium (K), temperature (T), humidity (H), pH level, and rainfall.
- **Data Preprocessing**: Clean and prepare the data for use in the machine learning models.
- **Model Training**: Train multiple models (e.g., KNN, Decision Trees, Random Forest) and evaluate their performance.
- **Prediction**: Use the trained models to predict suitable crops based on new environmental input data.
- **User Interaction**: Create a simple user interface where users (farmers) can input environmental data and receive crop recommendations.
- **Admin Panel**: Admins can update data and manage the training process.

3.3 Context Diagram (High Level)



4 Implementation Methodology

Step 1: **Data Collection** – Collect a dataset that includes environmental factors and their corresponding crops.

Step 2: **Data Preprocessing** – Handle missing values, encode categorical data, and normalize features.

Step 3: **Model Selection and Training** – Implement various machine learning models and evaluate their accuracy using cross-validation.

Step 4: **Prediction and Evaluation** – Use the best-performing model to make predictions and evaluate its performance on unseen data.

Step 5: **Deployment** – Build a user-friendly interface for farmers to input data and view predictions. Implement an admin panel to monitor system performance.

5 Technologies to be used

5.1 Software Platform

- Python (for machine learning and data processing)
- scikit-learn (for building machine learning models)
- Pandas and NumPy (for data handling)
- Matplotlib for histograms

5.2 Hardware Platform

- The system can run on standard desktop systems with a minimum of 4GB RAM and a processor capable of handling Python and machine learning tasks.

5.3 Tools, if any

- Jupyter Notebooks (for experimentation and testing models)
- Visual Studio Code or PyCharm (for code development)

6 Advantages of this Project

- **Informed Decision Making**: Helps farmers select the most suitable crops based on environmental conditions, improving yield and resource usage.
- **Increased Efficiency**: Reduces the need for trial and error in crop selection, saving time, money, and resources.
- **Sustainable Agriculture**: Promotes environmentally friendly farming practices by recommending crops that naturally thrive in local conditions, reducing dependency on chemicals.
- **Cost Reduction**: Optimizes resource use (water, fertilizer) and reduces wastage, ultimately lowering farming costs.
- **Easy to Use**: Accessible interface for farmers, allowing them to make data-driven decisions without needing technical expertise.
- **Improved Resource Management**: By recommending crops that are well-suited to the local environment, the system helps optimize the use of water, fertilizers, and pesticides, promoting better resource allocation and reducing environmental impact.
- **Scalable and Customizable**: The system can be expanded to cater to different geographical locations and specific crop types, allowing it to scale with agricultural practices across various regions. Customization can also include adding new features, such as pest management recommendations.
- **Data-Driven Insights**: By leveraging historical agricultural data, the system can provide insights into which crops perform well in specific areas, assisting farmers in selecting crops that have historically yielded high returns in similar conditions.
- **Helps in Climate Adaptation**: The system can assist farmers in adapting to changing climates by recommending crops that are resilient to extreme weather conditions, such as drought-resistant varieties in areas prone to water shortages.
- **Real-Time Crop Monitoring**: The system can integrate with crop monitoring technologies to provide real-time updates on crop health, growth progress, and environmental conditions, helping farmers take corrective actions early.

7 Assumptions, if any

None

8 Future Scope and further enhancement of the Project

- **Integration with IoT**: Real-time data from sensors could provide more accurate recommendations, adapting to current environmental conditions.
- **Mobile App**: A mobile version of the system will allow farmers to receive recommendations and input data while working in the field.
- **Weather Forecast Integration**: Using weather forecasts, the system can adjust recommendations based on upcoming climate conditions, ensuring timely decision-making.
- **Soil Health Data**: Incorporating soil health metrics will allow for even more accurate and tailored crop recommendations.
- **Advanced Machine Learning Models**: The system can be upgraded with deep learning for more sophisticated, data-driven predictions.
- **Collaboration with Research Institutes**: Partnerships with agricultural agencies can continuously improve the system with new research, ensuring its recommendations are always up-to-date.
- **Integration with Precision Agriculture Tools**: Incorporating GPS-based tools for precision farming could help farmers monitor field conditions at a more granular level. This can ensure that inputs like water, fertilizer, and pesticides are applied only where needed.
- **Automated Farm Machinery Integration**: The system could potentially be connected with automated machinery (e.g., robotic planters, harvesters) to execute the recommendations on the ground without manual intervention.
- **Market Price Predictions**: Integrating market trends and price predictions can help farmers decide which crops to grow based not only on environmental conditions but also on expected profitability. This would enhance the economic benefits of crop recommendation.
- **Collaborative Platform for Knowledge Sharing**: The system could include a feature for farmers to share their experiences and feedback about crop growth, pest outbreaks, and other local conditions, creating a community-driven knowledge base.
- **Predictive Analytics for Pest and Disease Management**: The system could be enhanced with predictive analytics to help forecast pest and disease outbreaks based on environmental factors, enabling farmers to take proactive measures before these issues arise.
- **Integration with Blockchain for Transparency**: For further transparency and traceability, integrating blockchain technology could help ensure the legitimacy of the crop recommendations, provide data security, and allow farmers to track the history of their farming practices.
- **Personalized Crop Advisory**: Over time, the system could offer more personalized crop advisory based on a farmer's past data and performance, building a tailored approach that improves with usage.

9 Project Repository Location

S#	Project Artifacts (softcopy)	Location (Mention Lab-ID, Server ID, Folder Name etc.)	Verified by Project Guide	Verified by Lab In-Charge
1.	Project Synopsis Report (Final Version)		Name and Signature	Name and Signature
2.	Project Progress updates		Name and Signature	Name and Signature
3.	Project Requirement specifications		Name and Signature	Name and Signature
4.	Project Report (Final Version)		Name and Signature	Name and Signature
5.	Test Repository		Name and Signature	Name and Signature
6.	Any other document, give details		Name and Signature	Name and Signature

10 Definitions, Acronyms, and Abbreviations

Abbreviation	Description
Machine Learning	Machine learning is a subset of artificial intelligence that focuses on enabling systems to learn from data and improve their performance without being explicitly programmed.
Sci-Kit Learn	Scikit-learn is an open-source Python library for machine learning and data modeling.
Pandas	Pandas is a Python library used for data manipulation and analysis.
Numpy	NumPy is a fundamental Python library for scientific computing.
Matplotlib	Matplotlib is a Python library used for creating static, animated, and interactive visualizations.
KNN	KNN is a supervised learning algorithm used for both classification and regression tasks
Decision Tree	Decision Tree is a flowchart-like structure that models decisions and their possible consequences, used for both classification and regression tasks.
Random Forest	Random forest is a popular machine learning algorithm that uses an ensemble of decision trees to make predictions.
Logistic Regression	Logistic regression is a machine learning algorithm used for binary classification, predicting the probability of a specific outcome based on input features.
Navie Bayes	Naive Bayes is a simple probabilistic classifier, based on Bayes' Theorem, that's widely used for classification tasks in machine learning.
SVM	SVM is a type of supervised learning algorithm used for both classification and regression

11 Conclusion

The **Crop Recommendation System** project represents a significant advancement in the application of machine learning and data science to modern agriculture. By utilizing environmental factors such as nitrogen, phosphorous, potassium, temperature, humidity, pH, and rainfall, this system provides farmers with tailored crop recommendations, empowering them to make data-driven decisions for better crop selection.

The system's ability to predict the best-suited crops for a given environment helps optimize resource use, improve crop yields, and reduce costs, promoting sustainable farming practices. Furthermore, its scalability allows for adaptation to various geographic locations and crop types, making it a versatile tool for farmers worldwide.

With the integration of real-time data from IoT sensors, weather forecasting, and mobile applications, the system can evolve to provide even more accurate, timely, and accessible recommendations, thus contributing to the broader goal of enhancing agricultural productivity and sustainability.

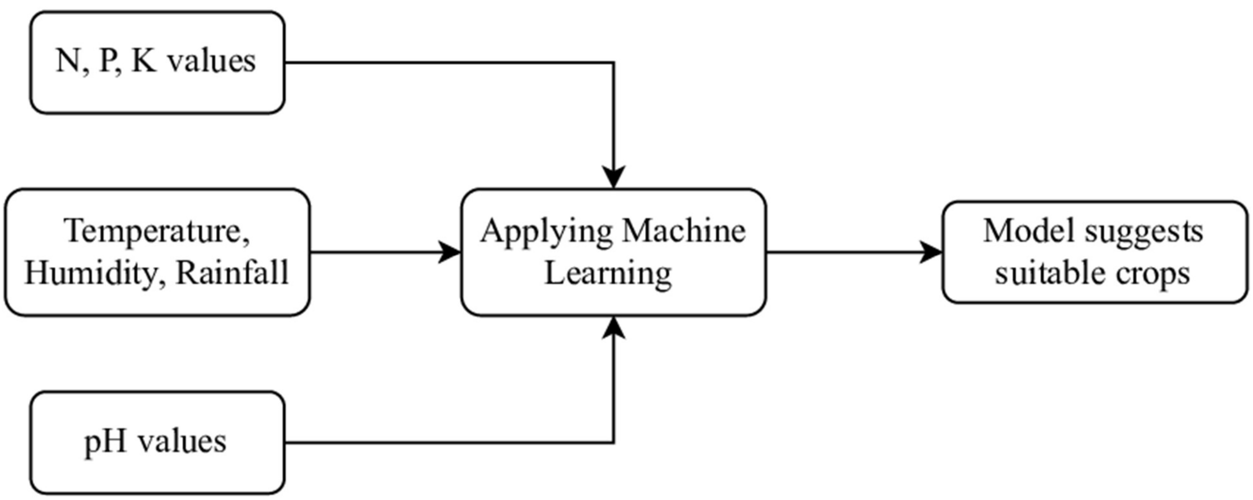
This project has the potential to revolutionize traditional farming methods, offering practical benefits that directly impact farmers' productivity and livelihoods. By adopting this system, farmers can better manage resources, reduce environmental impact, and adapt to the challenges posed by climate change.

As the system continues to improve and expand with future enhancements, it will play a pivotal role in ensuring the long-term sustainability of agriculture, fostering global food security, and supporting the continued growth of the agricultural sector.

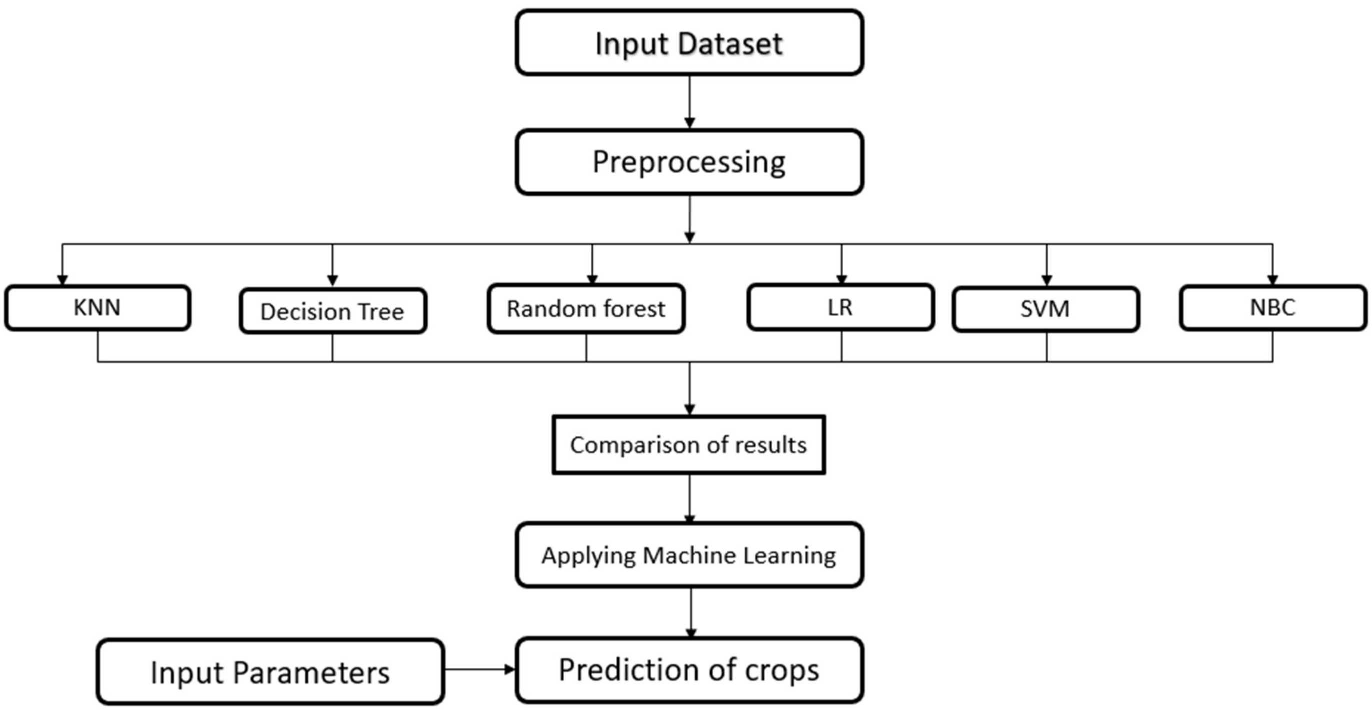
12 References

S#	Reference Details	Owner	Version	Date
1.	Project Synopsis	<Project Group ID>	1.0	26/04/2025
2.	Project Report	<Project Group ID>	1.0	26/04/2025
3.	Required Dataset	Kaggle		

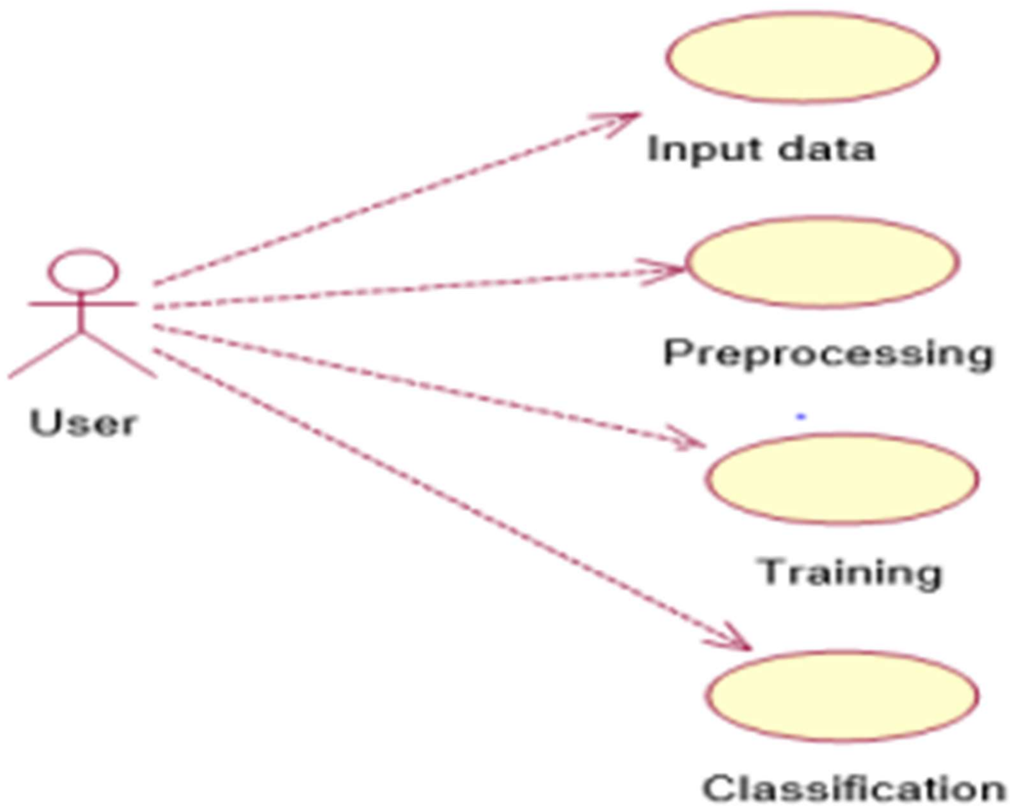
Annexure A
Data Flow Diagram (DFD)



Annexure B
Entity-Relationship Diagram (ERD)



Annexure C
Use-Case Diagram (UCD)



Annexure D
Data Dictionary (DD)

Environmental Features (Input Data)

Fields	Data type	Description
N	float	Nitrogen content in the soil (in kg per hectare)
P	float	Phosphorous content in the soil (in kg per hectare)
K	float	Potassium content in the soil (in kg per hectare)
Temperature	float	Average temperature of the region (in Celsius)
Humidity	float	Average humidity level of the region (in percentage)
pH	float	pH level of the soil
Rainfall	float	Average rainfall in the region (in mm)

Crop Data (Output Data)

Fields	Data type	Description
crop_id	Integer	Unique identifier for each crop
crop_name	String	Name of the recommended crop
recommended_for	String	Environmental conditions or regions where the crop is suitable

Annexure E
Screen Shots

Importing Python Libraries:

```
import pandas as pd
import numpy as np
import matplotlib
```
[27] ✓ 0.0s Python

Load Dataset:

```
cropdf = pd.read_csv("Crop_recommendation.csv")
cropdf.head()
```
[2] ✓ 0.0s Python

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

About Datset:

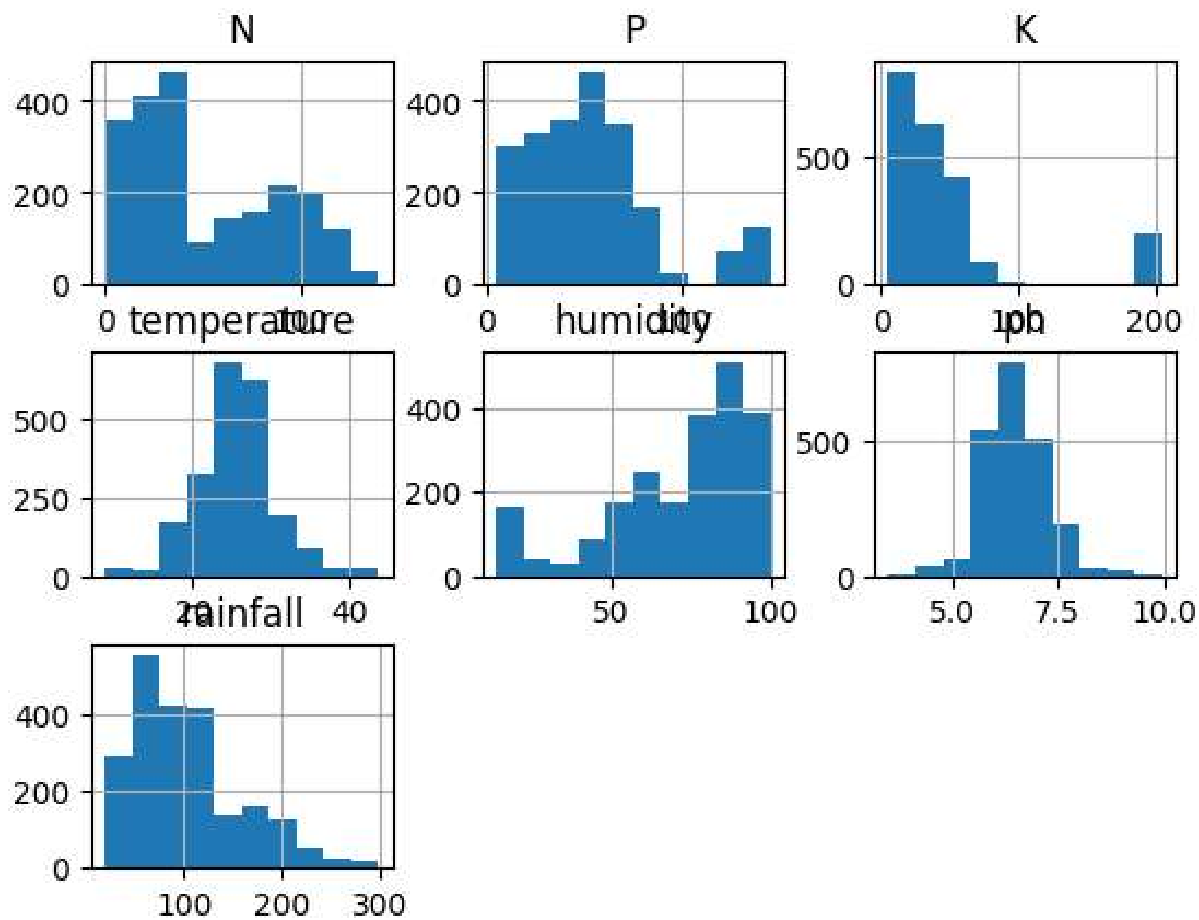
```
cropdf.info()
```
[3] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            2200 non-null   int64
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

Histograms:

```
cropdf.hist()
[4] ✓ 1.2s                                                          Python

... array([[<Axes: title={'center': 'N'}>, <Axes: title={'center': 'P'}>,
           <Axes: title={'center': 'K'}>],
          [<Axes: title={'center': 'temperature'}>,
           <Axes: title={'center': 'humidity'}>,
           <Axes: title={'center': 'ph'}>],
          [<Axes: title={'center': 'rainfall'}>, <Axes: >, <Axes: >]],
         dtype=object)
```



Separating Independent & Dependent Features:

```
X = cropdf.drop('label', axis=1)
y = cropdf['label']
```
[33] Python

```
X
```
[34] Python

	N	P	K	temperature	humidity	ph	rainfall
0	90	42	43	20.879744	82.002744	6.502985	202.935536
1	85	58	41	21.770462	80.319644	7.038096	226.655537
2	60	55	44	23.004459	82.320763	7.840207	263.964248
3	74	35	40	26.491096	80.158363	6.980401	242.864034
4	78	42	42	20.130175	81.604873	7.628473	262.717340
...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507
2196	99	15	27	27.417112	56.636362	6.086922	127.924610
2197	118	33	30	24.131797	67.225123	6.362608	173.322839
2198	117	32	34	26.272418	52.127394	6.758793	127.175293
2199	104	18	30	23.603016	60.396475	6.779833	140.937041

2200 rows × 7 columns

```
y
```
[35] Python

```
0         rice
1         rice
2         rice
3         rice
4         rice
         ...
2195    coffee
2196    coffee
2197    coffee
2198    coffee
2199    coffee
Name: label, Length: 2200, dtype: object
```

Split dataset for training and testing:

```
[36]
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.2,
shuffle = True, random_state = 0)
```

```
[37]
X_train
```

	N	P	K	temperature	humidity	ph	rainfall
1567	27	120	200	21.452787	90.745319	6.110219	116.703658
2031	81	36	38	23.765547	87.983299	6.334838	150.316615
2073	90	59	35	24.251335	89.864541	7.098228	175.174211
1374	81	16	45	26.904357	86.254262	6.727468	59.759800
279	47	80	77	17.182484	16.428918	7.561108	72.850173
...	...	...	...	...	...	...	...
1033	102	71	48	28.654563	79.286937	5.695268	102.463378
1731	34	68	51	27.347349	94.177567	6.687088	40.351531
763	35	64	15	28.474423	63.536045	6.500145	69.527441
835	39	65	23	25.434598	69.126134	7.685959	41.026829
1653	14	22	9	17.249446	91.137728	6.543192	112.509052

1760 rows × 7 columns

```
[38]
Y_train
```

```
1567         apple
2031          jute
2073          jute
1374    watermelon
279       chickpea
           ...
1033        banana
1731        papaya
763      blackgram
835         lentil
1653        orange
Name: label, Length: 1760, dtype: object
```

Building Machine Learning Models:

```python
def models(X_train, Y_train):
    # 1. K-Nearest Neighbors
    from sklearn.neighbors import KNeighborsClassifier
    knc = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
    knc.fit(X_train, Y_train)

    # 2. Decision Tree
    from sklearn.tree import DecisionTreeClassifier
    dtc = DecisionTreeClassifier(random_state=0, criterion="entropy")
    dtc.fit(X_train, Y_train)

    # 3. Logistic Regression
    from sklearn.linear_model import LogisticRegression
    lr = LogisticRegression(random_state=0)
    lr.fit(X_train, Y_train)

    # 4. Random Forest
    from sklearn.ensemble import RandomForestClassifier
    rfc = RandomForestClassifier(random_state=0, criterion="entropy", n_estimators=10)
    rfc.fit(X_train, Y_train)

    # 5. Support Vector Machine
    from sklearn.svm import SVC
    svc = SVC(random_state=0)
    svc.fit(X_train, Y_train)

    # 6. Naive Bayes Classifier
    from sklearn.naive_bayes import GaussianNB
    nbc = GaussianNB()
    nbc.fit(X_train, Y_train)

    print('[0] K-Nearest Neighbors accuracy:', knc.score(X_train, Y_train))
    print('[1] Decision Tree accuracy:', dtc.score(X_train, Y_train))
    print('[2] Logistic Regression accuracy:', lr.score(X_train, Y_train))
    print('[3] Random Forest accuracy:', rfc.score(X_train, Y_train))
    print('[4] Support Vector Machine accuracy:', svc.score(X_train, Y_train))
    print('[5] Naive Bayes Classifier accuracy:', nbc.score(X_train, Y_train))

    return knc, dtc, lr, rfc, svc, nbc
model = models(X_train,Y_train)
```

[39] Python

```
c:\Users\jains\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\linear_model\_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
[0] K-Nearest Neighbors accuracy: 0.9886363636363636
[1] Decision Tree accuracy: 1.0
[2] Logistic Regression accuracy: 0.9693181818181819
[3] Random Forest accuracy: 0.9994318181818181
[4] Support Vector Machine accuracy: 0.9767045454545454
[5] Naive Bayes Classifier accuracy: 0.9954545454545455
```

- Random Forest has the highest accuracy so we will predict new input by RandomForestClassifier

Applying Random Forest Model:

```python
from sklearn.ensemble import RandomForestClassifier
model1 = RandomForestClassifier()
```

[40] Python

```python
model1.fit(X_train, Y_train)
```

[41] Python

```
RandomForestClassifier
RandomForestClassifier()
```

```python
y_pred = model1.predict(X_test)
```

[42] Python

```python
def predict_new_data(N, P, K, T, H, PH, R):
    # Create a new DataFrame with the provided input features
    new_data = pd.DataFrame({
        'N': [N],
        'P': [P],
        'K': [K],
        'temperature': [T],
        'humidity': [H],
        'ph': [PH],
        'rainfall': [R]
    })
    # Predict using the model (assumes 'model1' is pre-trained and available)
    prediction = model1.predict(new_data)
    predicted_class = prediction[0]
    return predicted_class
```

[43] Python

Result:

```
N = str(input("Enter value for N (Nitrogen): "))
P = str(input("Enter value for P (Phosphorous): "))
K = str(input("Enter value for K (Potassium): "))
T = str(input("Enter value for temperature: "))
H = str(input("Enter value for humidity: "))
PH = str(input("Enter value for pH: "))
R = str(input("Enter value for rainfall: "))
```
[47] 19.8s Python

```
# Call the prediction function
predicted_class = predict_new_data(N, P, K, T, H, PH, R)
```
[48] 0.0s Python

```
# Print the prediction result
print(f"The predicted class for the input data is: {predicted_class}")
```
[49] 0.0s Python

```
The predicted class for the input data is: rice
```

CodeApto

Bengaluru, Karnataka
India

02 April, 2025
IN-BLR231-184

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Sahil Jain**, undergoing **CodeApto Empowerment Remote Internship Program,** from **CodeApto** has successfully completed her internship as per curriculum from **24th Aug, 2024 to 31st March, 2025**.
As a **Full Stack Data Science Development**, her role focused on an **interdisciplinary approach**, combining **Machine Learning** with **Python Programming** to analyze the requirements and develop enterprise grade applications.

To the best of our knowledge, she has proven to be both sincere and hardworking, consistently delivering satisfactory results during her tenure with us.

We wish her all the best for his future endeavors.

Yours truly,

Balwant Singh

Balwant Singh, Founder & CEO

CodeApto India Private Limited

Registered office - Jeevan Sai Meadows, 3rd Floor, Manjunath Layout, Bangalore - 560037