CROP PREDICITON USING MACHINE LEARNING

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATION

In

COMPUTER SCIENCE AND ENGINEERING (MCA)

Submitted By

CHINNALA ANUSHA **23UK1F0013**

Under the guidance of

Mr. RAJENDER REDDY NALLA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (MCA) VAAGDEVI ENGINEERING COLLEGE (AUTONOMOUS)

Affiliated to JNTUH, HYDERABAD , BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF COMPUTER SCIENCEAND ENGINEERING(MCA)

VAAGDEVI ENGINEERING COLLEGE (AUTONOMOUS)

CERTIFICATE OF COMPLETION PROJECT WORK REVIEW-I

This is to certify that the UG Project Phase-1 entitled "CROP PREDICITION USING MACHINE LEARNING" is being submitted by **CHINNALA ANUSHA (23UK1F0013)** in partial fulfillment of the requirements for the award of the degree of Master of computer application in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023-2024.

Project Guide
Mr. RAJENDER REDDY NALLA
(Assistant Professor)

HOD
Dr .R Naveenkumar
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr Syed Musthak Ahmed,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this PG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.R NaveenKumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the PG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the PG Project Phase-1 and for their support in completing the PG Project Phase-1.

We express heartfelt thanks to the guide, **Mr. K Ranjithkumar,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this PG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and out pouring their knowledge and experience throughout the thesis.

CHINNALA ANUSHA **(23UK1F0013)**

ABSTRACT

Farming has been the backbone of human civilization since the beginning of time. Farmers are responsible for feeding the world's population, and their efforts are critical in ensuring food security. However, selecting the best crop for a farm can be a daunting task, and farmers face several challenges in this regard. selecting crops is soil suitability. These challenges include different soil requirements based on Nitrogen, Phosphorus, Potassium content in the soil, different PH value of soil, variability in climate of a region based on humidity, rainfall, temperature of a particular region. Therefore, selecting the best crop for a farm is tedious job for farmers nowadays and it requires careful consideration of several factors.

Catering to all the challenges stated above, we have developed a model which can predict up to 21 Crops when given features based on soil content and climate-based factors of a particular region as input. This model can be used by farmers for predicting best crop for their farm in order to maximize their farm yield and also their profits through farming. This model does not ask for personalised data such as name, age, gender, religion, address, etc. You can use this web application anytime and the model will give the best crop based on the soil content and climate-oriented factors.

Table of Contents

1.INTRODUCTION

1.1 OVERVIEW

Crop Prediction is the process of using various techniques and technologies to forecast crop performance, yield, health, and other important parameters. It leverages data from diverse sources such as weather patterns, soil conditions, plant health, and historical crop data to make accurate and actionable predictions. These predictions assist farmers, agribusinesses, and policymakers in making informed decisions to optimize agricultural productivity and sustainability.

1.2 PURPOSE

1. **Optimizing Crop Yields**: By predicting the best times for planting, irrigation, and harvesting, crop prediction helps farmers maximize their yields.
2. **Soil Health Maintenance:** Predictive models can recommend crop rotations and other practices that maintain or improve soil health, ensuring long-term agricultural sustainability.
3. **Stable Food Supply:** By predicting crop yields and potential shortages, crop prediction contributes to maintaining a stable food supply, reducing the risk of food scarcity.
4. **Education and Training:** Providing training and resources to farmers on how to use predictive tools effectively enhances their capabilities and confidence in managing their farms.
5. **Government Policies:** Crop prediction data assists governments in formulating agricultural policies, subsidies, and support programs that are based on accurate and current information.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEMS

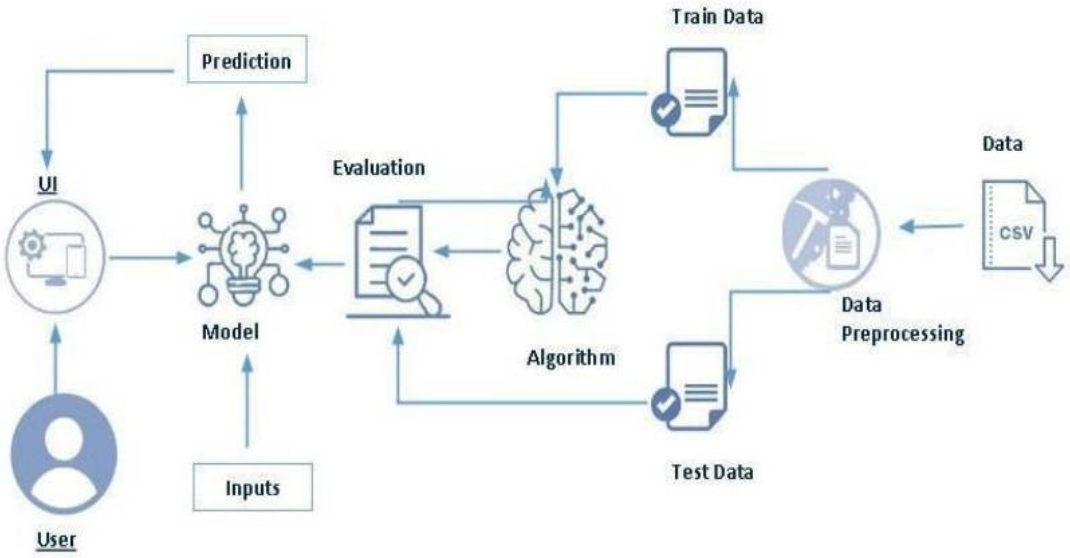
- **Climate Variability:** Unpredictable weather patterns and extreme events (droughts, floods) make accurate crop prediction difficult
- **Soil Health:** Soil degradation and nutrient depletion impact crop yields.
- **Complexity of Agricultural Systems**: Crop prediction models need to account for variables like weather, soil, crop type, management practices, and socio-economic factors.
- **Policy and Economic Factors**: Changes in agricultural policies, market conditions, and economic factors influence crop production. Predictive models need to consider these external factors for accurate forecasting.
- **Technology Adoption**: Resistance to adopting new technologies among farmers due to cost or lack of knowledge

2.1PURPOSED SOLUTIONS

- **Advanced Data Analytics and AI**: Utilizing machine learning and AI for more accurate and robust prediction models. Leveraging big data analytics to process and analyze large datasets.
- **Remote Sensing and GIS**: Using satellite imagery and GIS technologies to monitor and predict crop health and yields. Integrating remote sensing data with ground-based observations.
- **IoT and Precision Agriculture**: Deploying IoT sensors for real-time monitoring of soil moisture, temperature, and other critical parameters. Implementing precision agriculture techniques to optimize inputs and improve crop management.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Colab:** Google Colab will serve as the development and execution environment for your predictive modeling, data preprocessing, and model trainingtasks. It provides a cloud-based. Jupiter Notebook environment with access to Python libraries and hardware acceleration.
- **Dataset (CSV File):** The dataset in CSV format is essential for training and testingyour predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.
- **Data Preprocessing Tools:** Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.
- **Feature Selection/Drop:** Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

- **Model Training Tools:** Machine learning libraries such as Scikit-learn, TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the Crop Prediction task.
- **Model Accuracy Evaluation:** After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assessthe model's predictive capabilities. You'll measure the model's ability to predict CROP categories based on historical data.
- **UI Based on Flask Environment:** Flask, a Python web framework, will be used todevelop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view Crop Predictions, health information, and recommended precautions.
- Google Colab will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selectionwill optimize the model. Finally, model accuracy evaluation will confirm thesystem's predictive capabilities, allowing users to rely on the Crop predictions and associated health information.

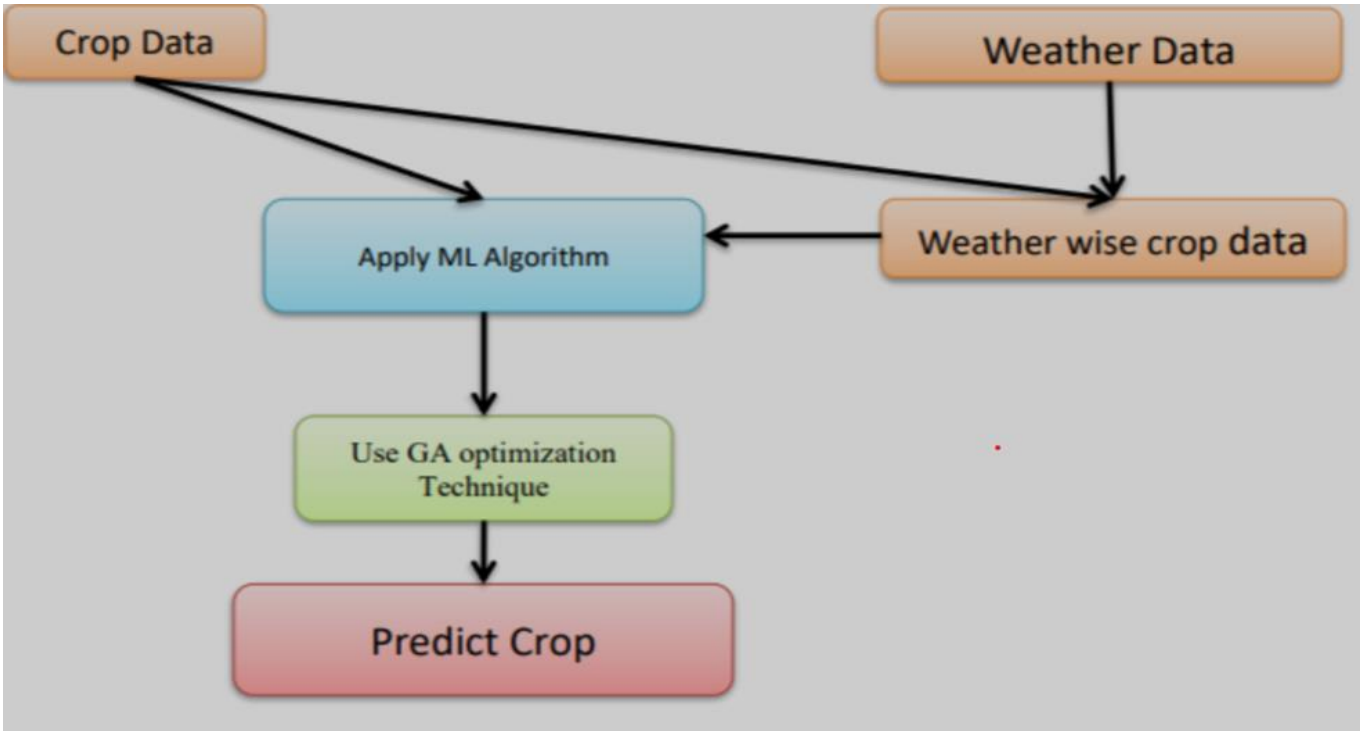
4. EXPERIMENTAL INVESTIGATION

The dataset consists of a single CSV file. This dataset is mainly made concerning Indian climatic conditions. There are, in total, seven input features and one output feature. The seven input features are as follows:

1. **N** - Ratio of Nitrogen content in the soil.
2. **P** - Ratio of Phosphorous content in the soil.
3. **K-** Ratio of Potassium content in the soil.
4. **Temperature** - The temperature in degrees Celsius.
5. **Humidity** - Relative humidity in %.
6. **Rainfall** - Rainfall in mm.
7. **Ph**- Ph value of the soil.

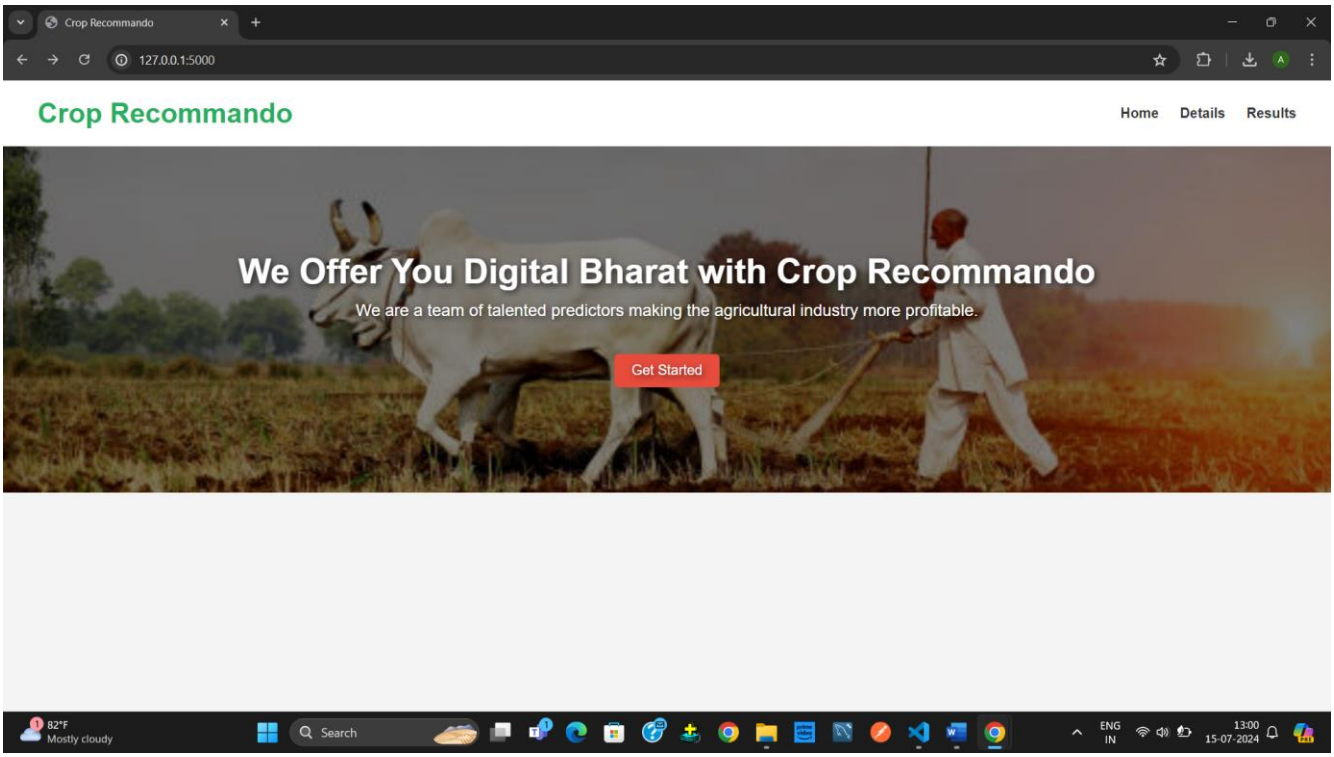
For the dataset we selected, it consists of more than the columns we want to predict it .So, we have chosen the feature drop it contains the columns that we are going to predict the Crop value.

5.FLOWCHART

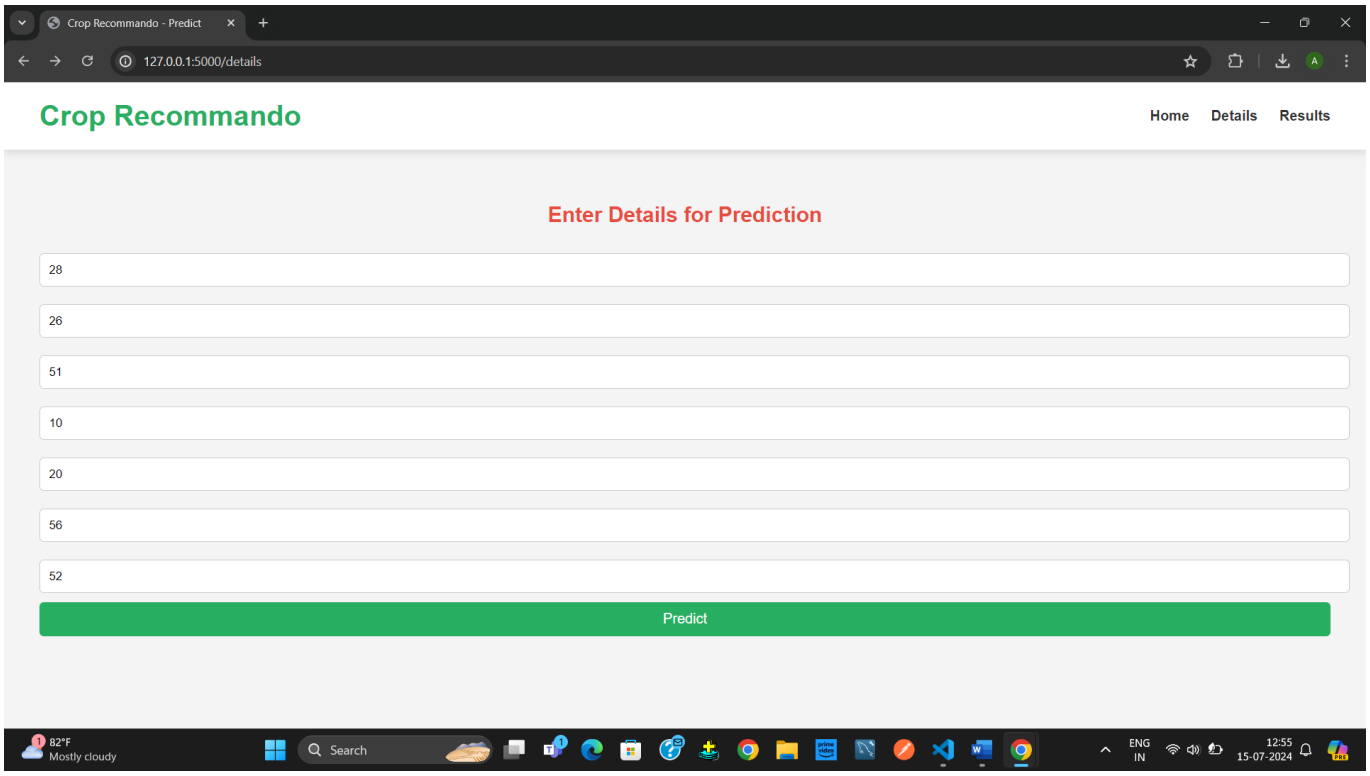

Crop Data
Weather Data
Apply ML Algorithm
Weather wise crop data
Use GA optimization Technique
Predict Crop

6.RESULT

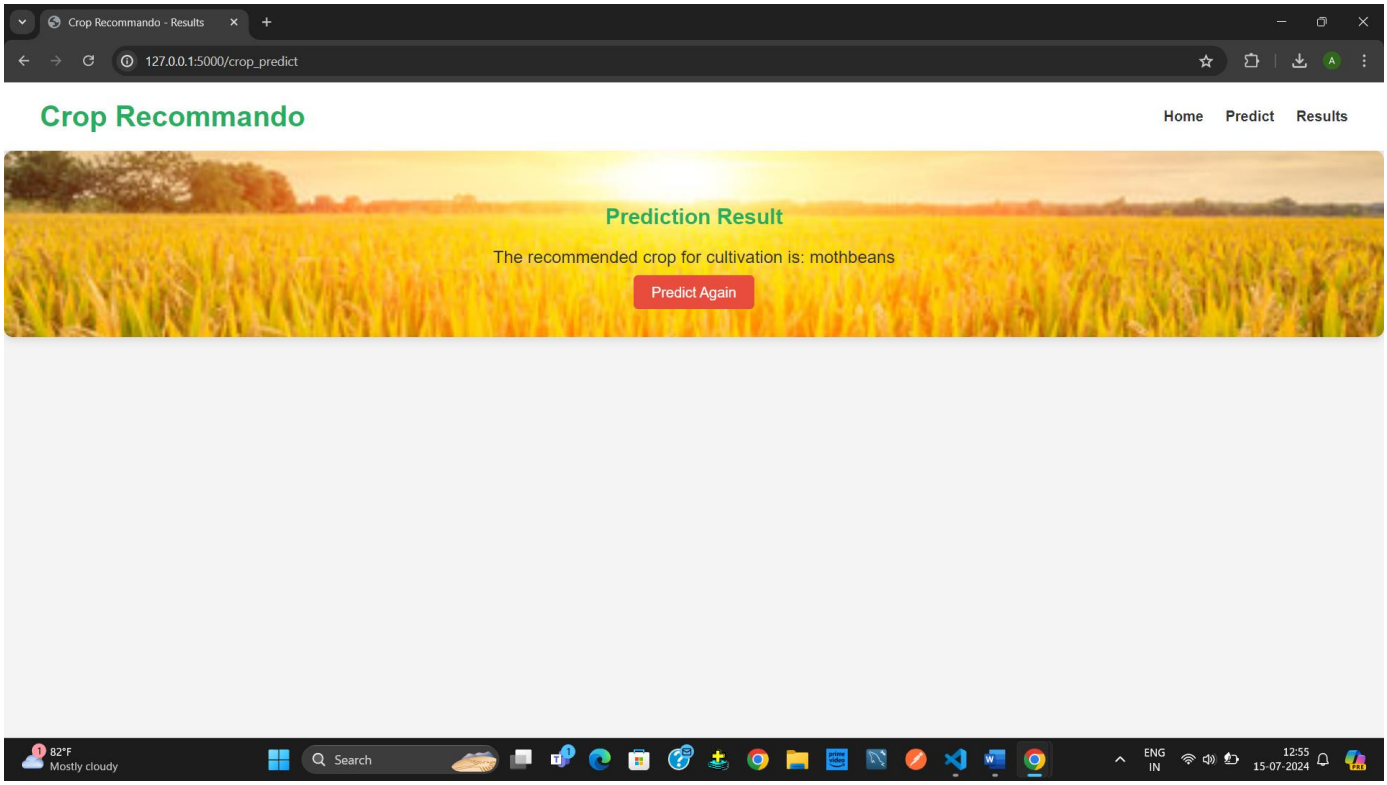
HOME PAGE



DETAILS


Crop Recommando - Predict
127.0.0.1:5000/details
Crop Recommando
Home
Details
Results
Enter Details for Prediction
28
26
51
10
20
56
52
Predict
82°F
Mostly cloudy
Search
ENG
IN
12:55
15-07-2024

RESULT


Crop Recommando - Results
127.0.0.1:5000/crop_predict
Crop Recommando
Home
Predict
Results
Prediction Result
The recommended crop for cultivation is: mothbeans
Predict Again
82°F
Mostly cloudy
Search
ENG
IN
12:55
15-07-2024

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Increased Crop Yield:** Predicting the best times for planting, watering, and harvesting can optimize crop growth and yield, leading to higher productivity.
2. **Resource Optimization:** Helps in the efficient use of resources like water, fertilizers, and pesticides, reducing waste and cost.
3. **Risk Management:** Farmers can anticipate adverse weather conditions, pest outbreaks, and other risks, allowing for proactive measures to mitigate losses.
4. **Economic Benefits:** Better crop management can lead to increased profitability for farmers through improved market timing and reduced crop losses.
5. **Sustainable Agriculture:** Encourages practices that are more environmentally friendly by optimizing inputs and reducing the negative impacts of farming.
6. **Market Planning:** Helps farmers and agribusinesses plan for market demands, reducing the chances of overproduction or underproduction.

DISADVANTAGES:

1. **High Initial Costs:** Implementing crop prediction technologies can be expensive, requiring investments in sensors, software, and training.
2. **Data Dependency:** Accurate predictions rely on high-quality data, which may not be available or accessible for all farmers, particularly in remote or underdeveloped areas.
3. **Technology Integration:** Farmers need to integrate new technologies with traditional farming practices, which can be challenging and require significant adjustments.
4. **Technical Expertise:** Requires a certain level of technical knowledge and expertise, which may not be available to all farmers, necessitating additional training.
5. **Maintenance and Upkeep:** The technology and equipment used for crop prediction require regular maintenance, which can be time-consuming and costly.

8.APPLICATIONS

1. **Yield Prediction:** Estimating the potential yield of crops before harvest helps farmers and agribusinesses plan for storage, transportation, and market sales.
2. **Irrigation Management:** Optimizing irrigation schedules based on crop water requirements and weather predictions to ensure efficient water use and prevent over- or under-watering
3. **Soil Health Monitoring:** Assessing soil conditions and predicting nutrient needs to inform fertilizer application, ensuring crops receive the necessary nutrients for optimal growth.
4. **Harvest Timing:** Determining the best time to harvest crops to maximize yield and quality, reducing losses due to premature or delayed harvesting.
5. **Market Forecasting:** Predicting crop supply and demand trends to inform pricing strategies and market positioning, helping farmers and traders make informed decisions.

9.CONCLUSION

- In conclusion, Crop prediction using machine learning represents a transformative approach to modern agriculture. By leveraging vast amounts of data from diverse sources such as weather patterns, soil conditions, historical yields, and satellite imagery, machine learning models can provide accurate and timely predictions of crop yields. This technology enables farmers to make informed decisions, optimize resource use, and improve crop management practices, ultimately leading to increased productivity and sustainability.
- Machine learning-based crop prediction holds immense potential for revolutionizing agriculture. By addressing existing challenges and harnessing technological advancements, this approach can lead to more resilient, efficient, and sustainable agricultural practices, benefiting farmers and the broader food production system

10.FUTURE SCOPE

Future Scope of the Crop Prediction Using Machine Learning:

1.**Enhanced Precision Agriculture:**

- **AI and Machine Learning:** Advanced AI algorithms will further refine predictive models, providing highly accurate and localized predictions.
- **Integration with IoT:** More sophisticated IoT devices will collect real-time data from the field, improving the accuracy of predictions and enabling immediate adjustments to farming practices.

2. **Remote Sensing and Satellite Technology:**

- **High-Resolution Imaging:** Improved satellite and drone technology will provide higher resolution images, enabling more detailed analysis of crop health and growth.
- **Global Monitoring:** Enhanced global monitoring capabilities will allow for better prediction of crop yields and health on a large scale, aiding in food security planning.

3. **Big Data Analytics:**

- **Data Fusion:** Combining diverse data sources, such as weather data, soil health data, and historical crop performance, to create more comprehensive and accurate prediction models.
- **Real-Time Analytics**: Leveraging big data analytics to process and analyze large volumes of data in real time, providing immediate insights and recommendations to farmers.

4. **Blockchain Technology:**

- **Data Security:** Using blockchain to secure data and ensure its integrity, making it easier to share and verify data among stakeholders.
- **Supply Chain Transparency:** Enhancing traceability and transparency in the agricultural supply chain, ensuring that crop predictions are used effectively and efficiently.

5**. Collaborative Platforms:**

- **Farmer Networks:** Creation of collaborative platforms where farmers can share data and insights, improving the overall accuracy of predictive models through collective intelligence.
- **Research Partnerships:** Increased collaboration between research institutions.

11.APPENDIX

Model building:

1) Dataset
2) Google colab and VS code Application Building
    1. HTML file (Index file, Details file ,Predict file)
    2. Appy.py file

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Crop Recommando</title>
    <style>
        /* General styles */
        body {
            font-family: 'Arial', sans-serif;
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            background-color: #f4f4f4;
            color: #333;
        }

        .header {
            display: flex;
            justify-content: space-between;
            align-items: center;
            background-color: #fff;
            padding: 20px 40px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }

        .header h1 {
            margin: 0;
            color: #27ae60;
        }
```

```
        .header nav a {
            margin-left: 20px;
            text-decoration: none;
            color: #333;
            font-weight: bold;
        }

        .header nav a:hover {
            color: #27ae60;
        }

        .hero {
            position: relative;
            height: 400px;
            text-align: center;
            color: #fff;
           background: url('/static/images/hero-image.jpg') no-repeat center center/cover;
            background-size: cover;
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
            text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.7);
        }


        .hero::before {
            content: "";
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background-color: rgba(0, 0, 0, 0.5); /* Adjust opacity as needed */
            z-index: 1;
        }

.hero h2 {
            font-size: 2.5em;
            margin: 0;
            z-index: 2;
        }

        .hero p {
            font-size: 1.2em;
```

```
            margin-top: 10px;
            z-index: 2;
        }

        .hero .btn {
            margin-top: 20px;
            padding: 10px 20px;
            font-size: 1em;
            color: #fff;
            background-color: #e74c3c;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            text-decoration: none;
            z-index: 2;
        }

        .hero .btn:hover {
            background-color: #c0392b;
        }
    </style>
</head>
<body>
    <header class="header">
        <h1>Crop Recommando</h1>
        <nav>
            <a href="/">Home</a>
            <a href="/details">Details</a>
            <a href="/crop_predict">Results</a>
        </nav>
    </header>
<section class="hero">
        <h2>We Offer You Digital Bharat with Crop Recommando</h2>
         <p>We are a team of talented predictors making the agricultural industry more profitable.</p>
        <a href="/details" class="btn">Get Started</a>
    </section>
</body></html>
```

DETAILS. HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Crop Recommando - Predict</title>
    <style>
        /* General styles */
        body {
            font-family: 'Arial', sans-serif;
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            background-color: #f4f4f4;
            color: #333;
        }

        .header {
            display: flex;
            justify-content: space-between;
            align-items: center;
            background-color: #fff;
            padding: 20px 40px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }

        .header h1 {
            margin: 0;
            color: #27ae60;
        }
.header nav a {
            margin-left: 20px;
            text-decoration: none;
            color: #333;
            font-weight: bold;
        }

        .header nav a:hover {
            color: #27ae60;
        }

        .form-container {
            padding: 40px;
            text-align: center;
        }


        .form-container h2 {
            color: #e74c3c;
```

```
            margin-bottom: 20px;
        }

        .form-container input {
            width: 100%;
            padding: 10px;
            margin: 10px 0;
            border: 1px solid #ccc;
            border-radius: 5px;
        }

        .form-container .btn {
            width: 100%;
            padding: 10px;
            font-size: 1em;
            color: #fff;
            background-color: #27ae60;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }

        .form-container .btn:hover {
            background-color: #219150;
        }
    </style>
</head>
<body>
    <header class="header">
        <h1>Crop Recommando</h1>
        <nav>
            <a href="/">Home</a>
            <a href="/details">Details</a>
            <a href="/crop_predict">Results</a>
        </nav>
    </header>
    <section class="form-container">
        <h2>Enter Details for Prediction</h2>
        <form action="/crop_predict" method="post">
            <input type="number" step="any" name="N" placeholder="N value" required>
            <input type="number" step="any" name="P" placeholder="P value" required>
            <input type="number" step="any" name="K" placeholder="K value" required>
             <input type="number" step="any" name="temperature" placeholder="Temperature"
required>
```

```
                <input  type="number"  step="any"  name="humidity"  placeholder="Humidity"
required>
            <input type="number" step="any" name="ph" placeholder="pH value" required>
                <input  type="number"  step="any"  name="rainfall"  placeholder="Rainfall"
required>
            <button type="submit" class="btn">Predict</button>
        </form>
    </section>
</body>
</html>
```

PREDICT. HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Crop Recommando - Results</title>
    <style>
        /* General styles */
        body {
font-family: 'Arial', sans-serif;
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            background-color: #f4f4f4;
            color: #333;
        }

        .header {
            display: flex;
            justify-content: space-between;
            align-items: center;
            background-color: #fff;
            padding: 20px 40px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }

        .header h1 {
            margin: 0;
            color: #27ae60;
        }
```

```
        .header nav a {
            margin-left: 20px;
            text-decoration: none;
            color: #333;
            font-weight: bold;
        }

        .header nav a:hover {
            color: #27ae60;
        }

        .result-container {
            padding: 40px;
            text-align: center;
               background: url('/static/images/result-background.jpg') no-repeat center
center/cover;
             /* Replace 'path_to_your_background_image.jpg' with the actual path to your
image */
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
.result-container h2 {
            color: #27ae60;
            margin-bottom: 20px;
        }

        .result-container p {
            font-size: 1.2em;
            margin-top: 10px;
        }

        .result-container .btn {
            margin-top: 20px;
            padding: 10px 20px;
            font-size: 1em;
            color: #fff;
            background-color: #e74c3c;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            text-decoration: none;
        }

        .result-container .btn:hover {
            background-color: #c0392b;
```

```
        }
    </style>
</head>
<body>
    <header class="header">
        <h1>Crop Recommando</h1>
        <nav>
            <a href="/">Home</a>
            <a href="/details">Predict</a>
            <a href="/crop_predict">Results</a>
        </nav>
    </header>
    <section class="result-container">
        <h2>Prediction Result</h2>
        {% if crop %}
            <p>The recommended crop for cultivation is: {{ crop }}</p>
{% else %}
            <p>No prediction available.</p>
        {% endif %}
        <a href="/details" class="btn">Predict Again</a>
    </section>
</body>
</html>
```

APP. PY:

```
from flask import Flask, render_template, request
import numpy as np
import pickle

model=pickle.load(open('model.pkl','rb'))

app = Flask(__name__, static_url_path='/static')

# Your other routes and functions

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/details')
def pred():
    return render_template('details.html')
```

```python
@app.route('/crop_predict', methods=['POST'])
def crop_predict():
    if not model:
        return "Model is not loaded properly. Please check the model file."

    # Get the input data from the form
    N = float(request.form['N'])
    P = float(request.form['P'])
    K = float(request.form['K'])
    temperature = float(request.form['temperature'])
    humidity = float(request.form['humidity'])
    ph = float(request.form['ph'])
    rainfall = float(request.form['rainfall'])

# Make a prediction using the loaded model
    prediction = model.predict([[N, P, K, temperature, humidity, ph, rainfall]])
    crop = prediction[0]

    return render_template('crop_predict.html', crop=crop)

# Run the Flask application
if __name__ == '__main__':
    app.run(debug=True)
```

CODE SNIPPETS

MODEL BUILDING:

Importing the Libraries

```
[9] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import numpy as np
    from sklearn.metrics import confusion_matrix, classification_report
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.svm import SVC
    from sklearn.preprocessing import StandardScaler
    from sklearn.model_selection import train_test_split
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.neighbors import KNeighborsClassifier
    import pickle
```

```
#reading dataset
crop_df = pd.read_csv("/content/Crop_recommendation.csv")
crop_df
```

	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
...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

2200 rows × 8 columns

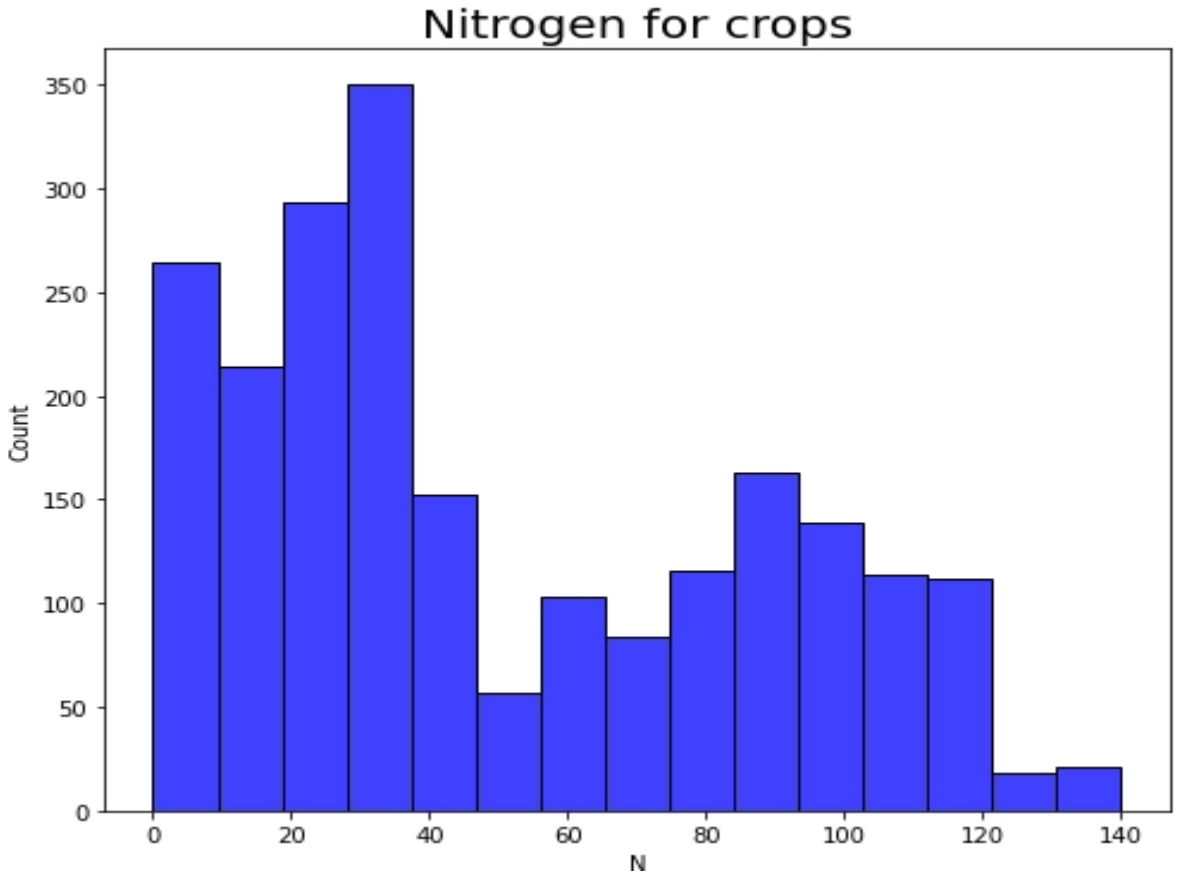
```
[ ] crop_df.shape

(2200, 8)
```

Data Preparation

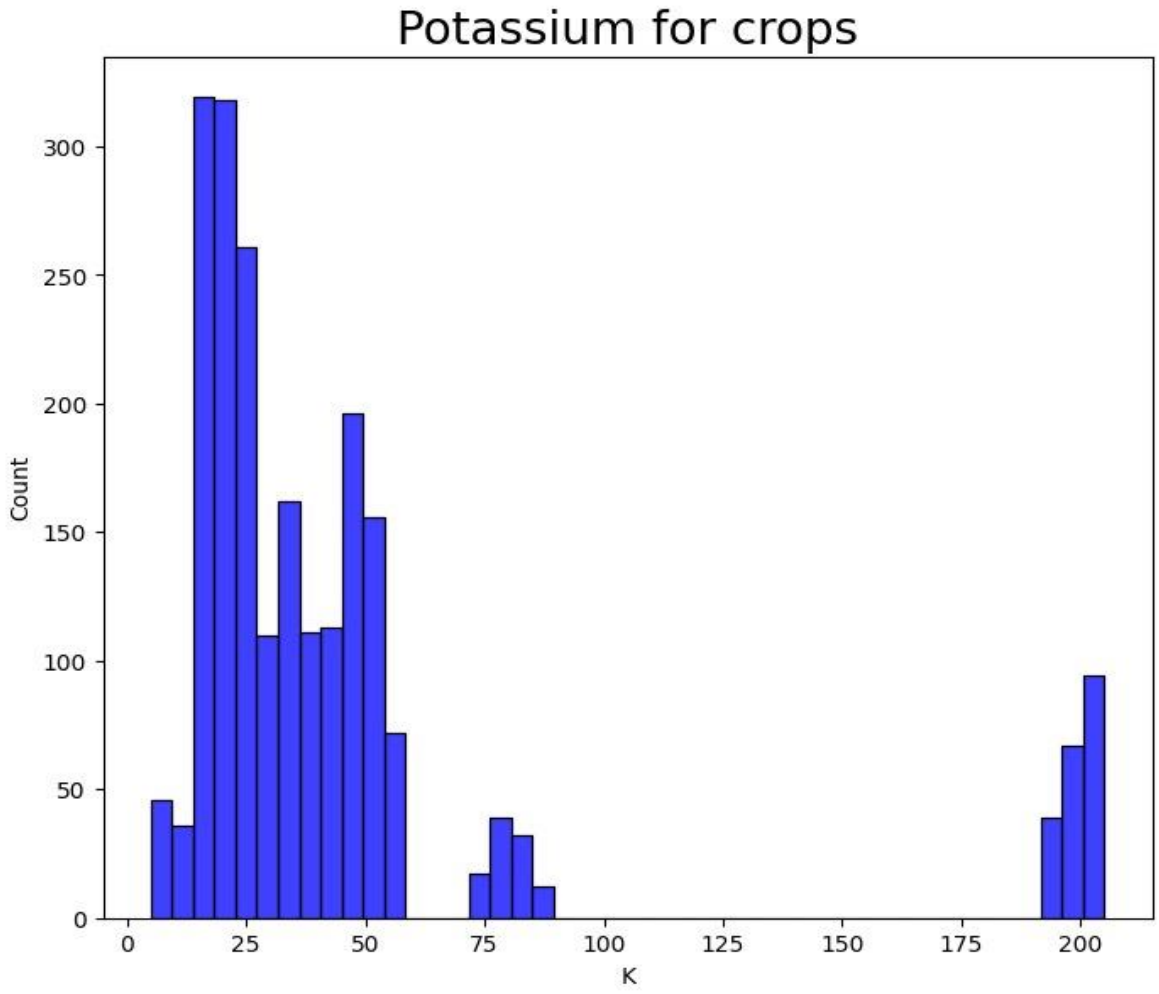
```
#Check null values are there or not
crop_df.isnull().sum()

N            0
P            0
K            0
temperature  0
humidity     0
ph           0
rainfall     0
label        0
dtype: int64
```

Visualisations

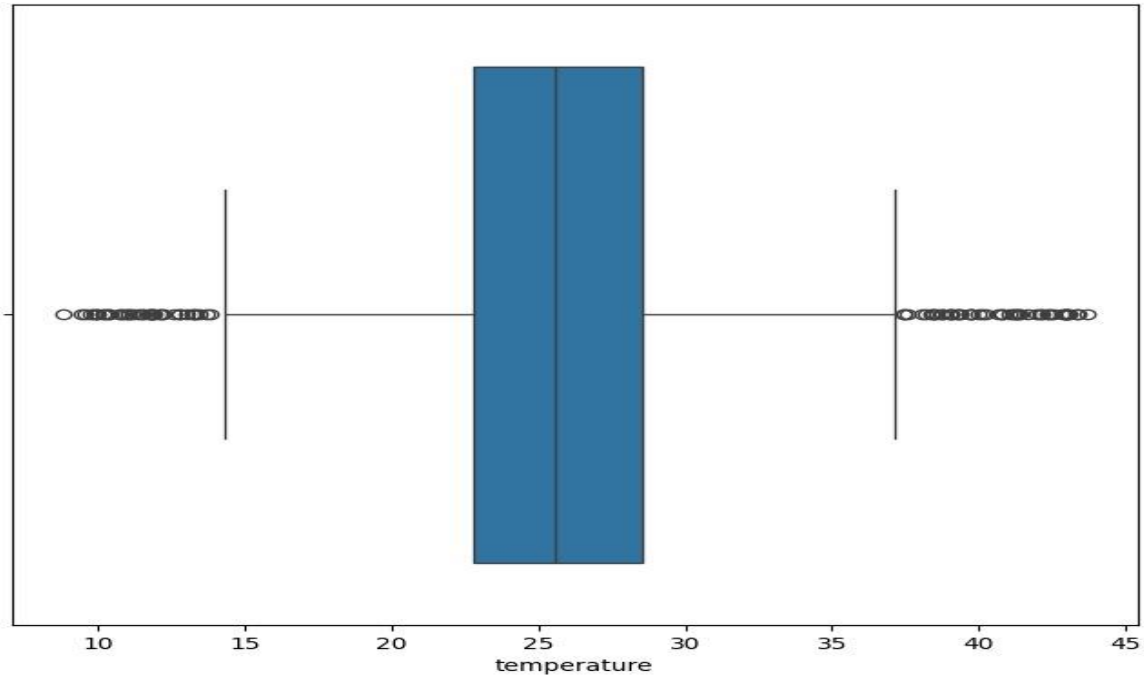
```
#Nitrogen
plt.figure(figsize=(8,7))
sns.histplot(x='N', data=crop_df, color = 'b')
plt.title("Nitogen for crops", {'fontsize': 20})
plt.show()
```



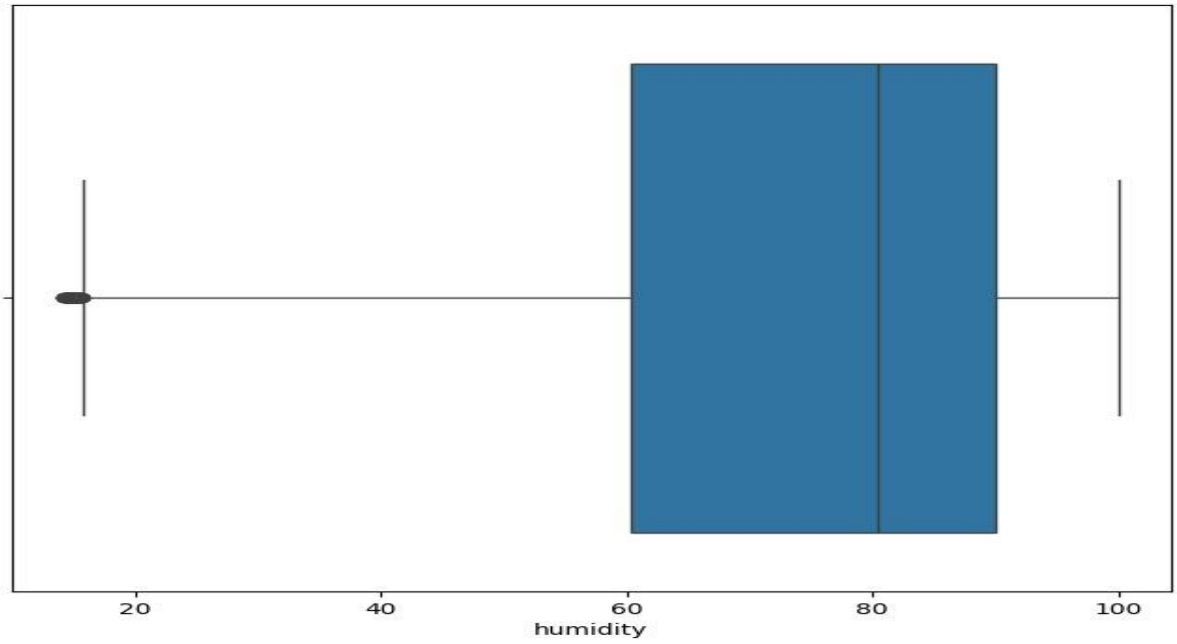

```
#Potassium
plt.figure(figsize=(8,7))
sns.histplot(x='K', data=crop_df, color = 'b')
plt.title("Potassium for crops", {'fontsize': 20})
plt.show()
```



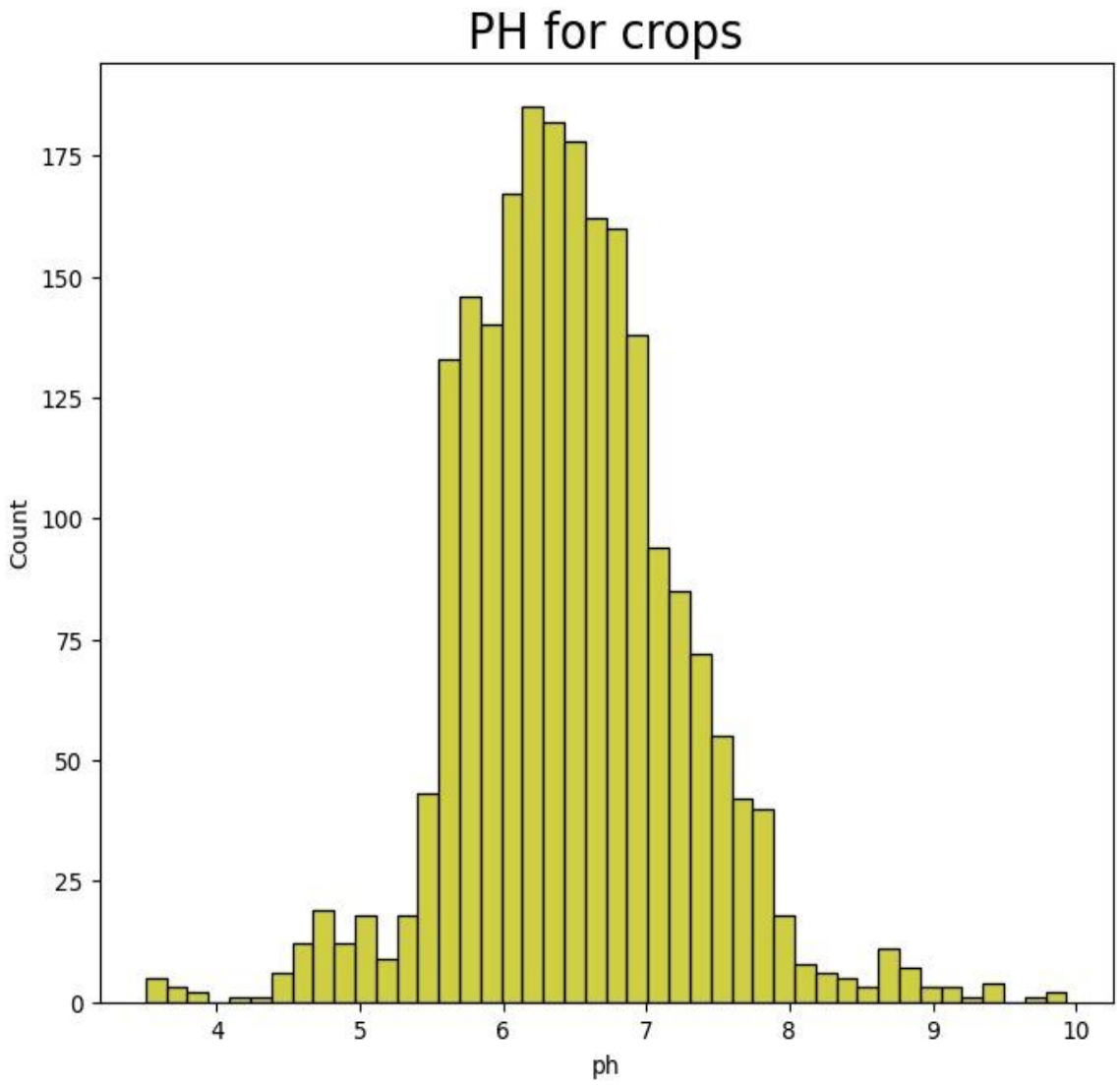

```
#Temperature
plt.figure(figsize=(8,7))
sns.boxplot(x= crop_df.temperature)
plt.show()
```



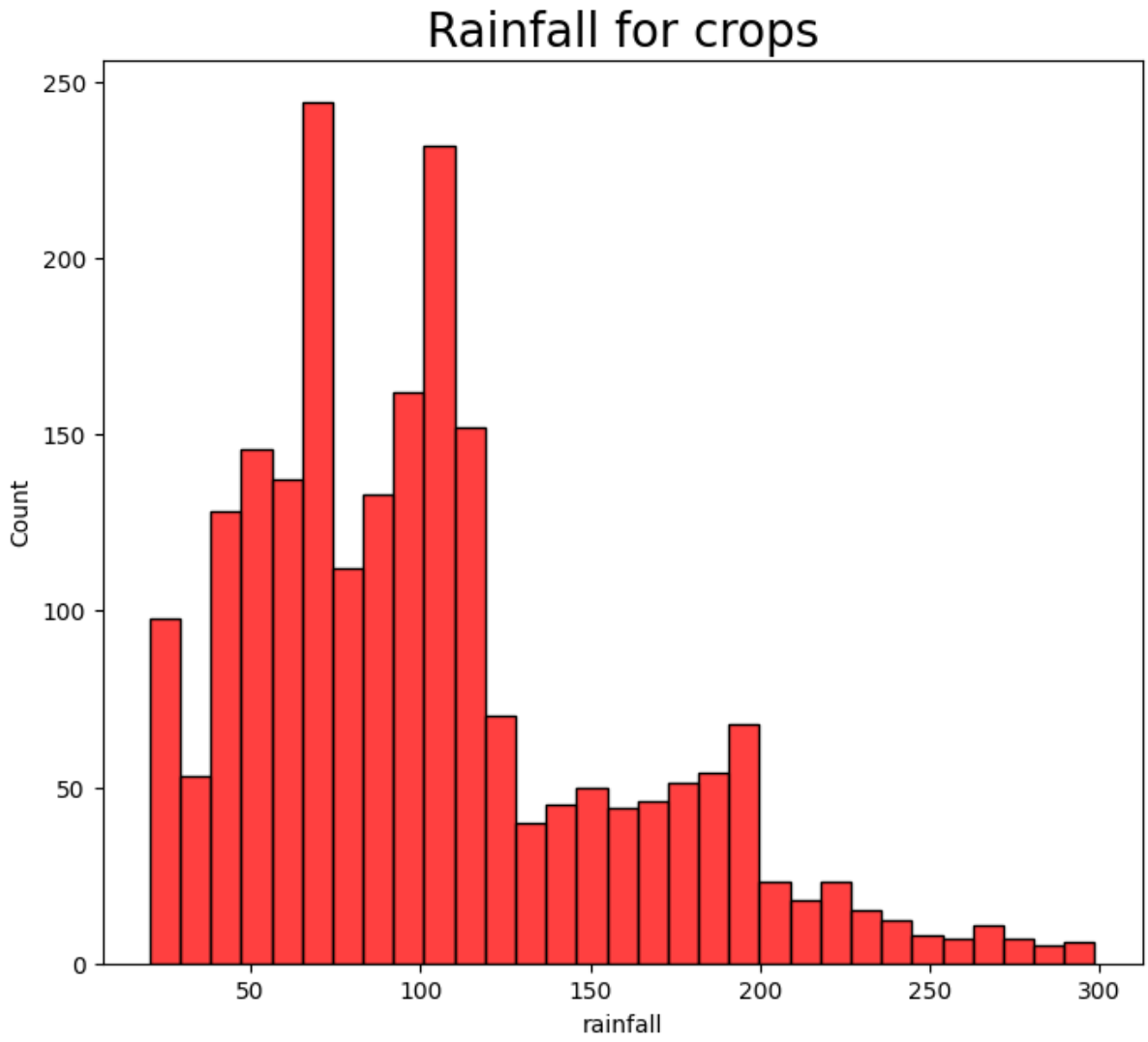

```
#Humidity
plt.figure(figsize=(8,7))
sns.boxplot(x= crop_df.humidity)
plt.show()
```



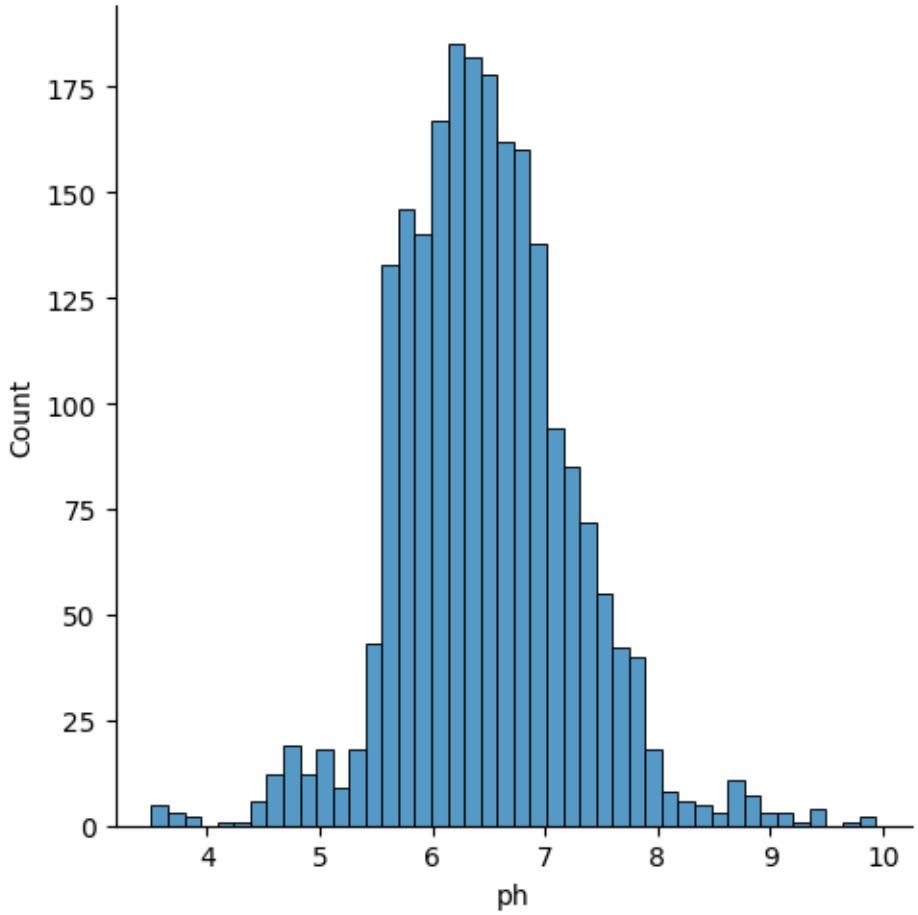
```
#PH
plt.figure(figsize=(8,7))
sns.histplot(x='ph', data=crop_df, color = 'b')
plt.title("PH for crops", {'fontsize': 20})
plt.show()
```



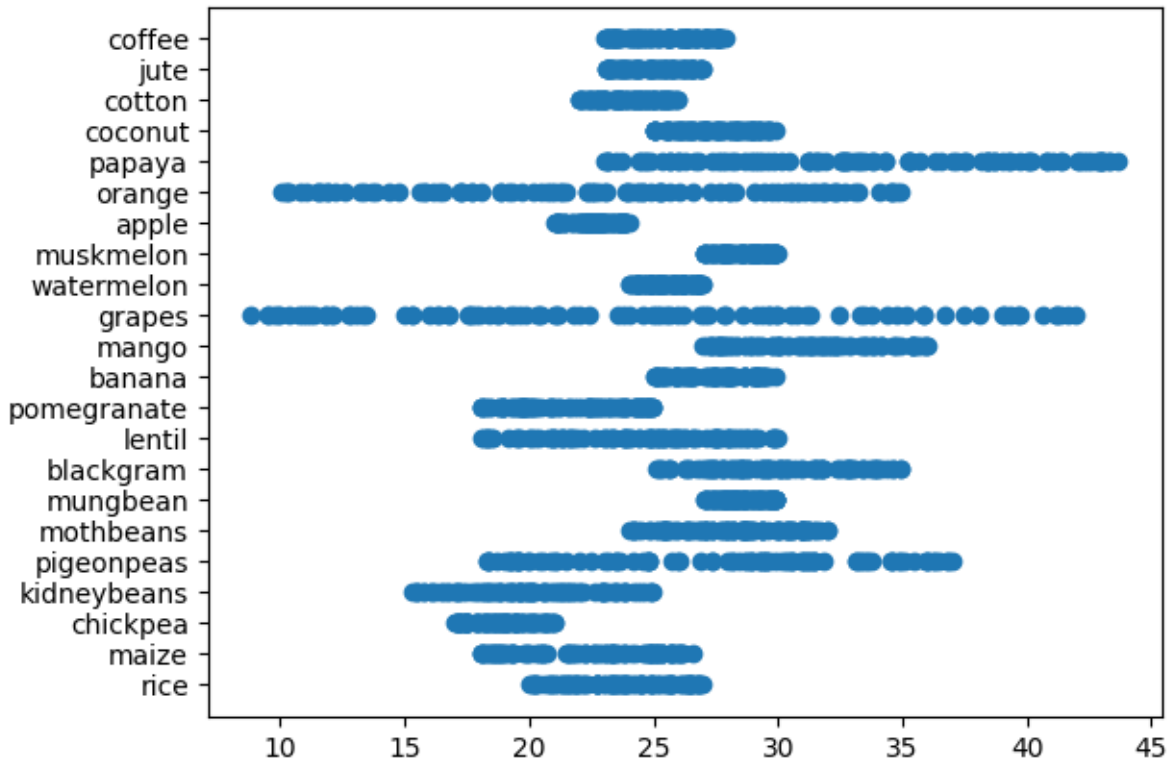

```
#Rainfall
plt.figure(figsize=(8,7))
sns.histplot(x='rainfall', data=crop_df, color = 'r')
plt.title("Rainfall for crops", {'fontsize': 20})
plt.show()
```




```
#Displot
sns.displot(crop_df['ph'])
plt.show()
```

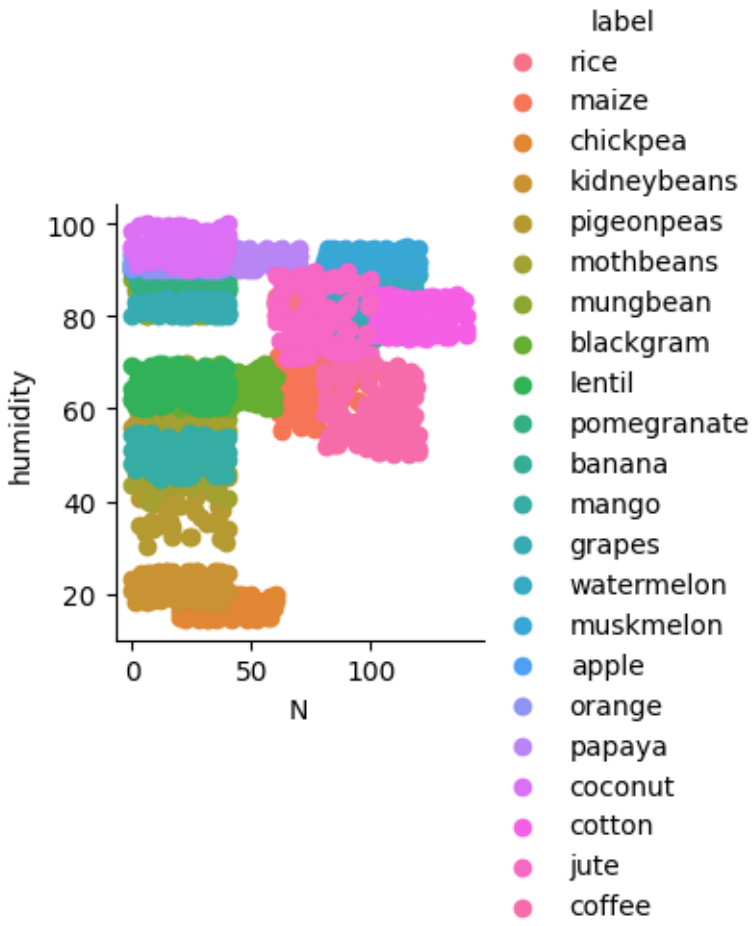



```
[ ]  #Scatterplot
     plt.scatter(crop_df["temperature"],crop_df["label"])
     plt.show()
```




```
#FacetGrid
sns.FacetGrid(crop_df,hue="label").map(plt.scatter,"N","humidity").add_legend()
```

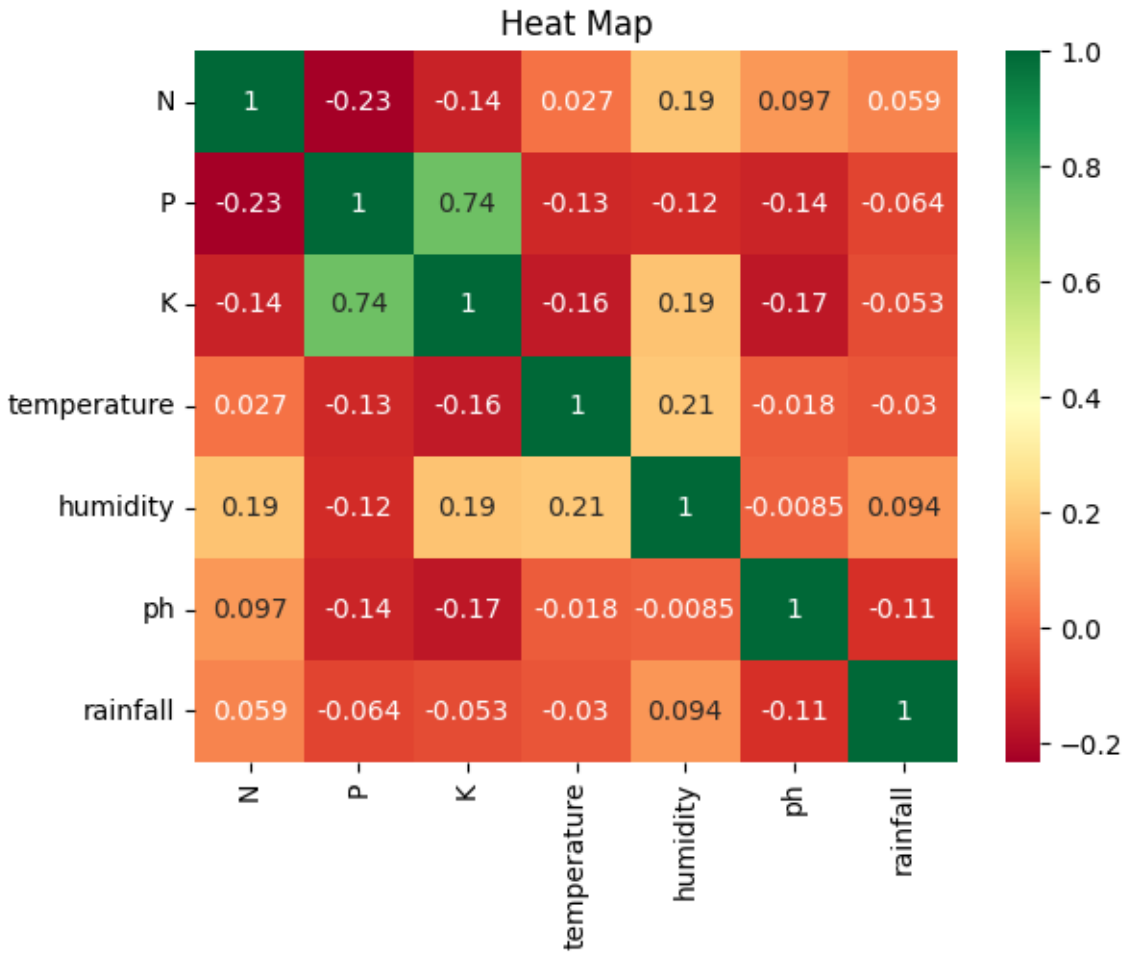
```
<seaborn.axisgrid.FacetGrid at 0x7ee7ea3ead70>
```



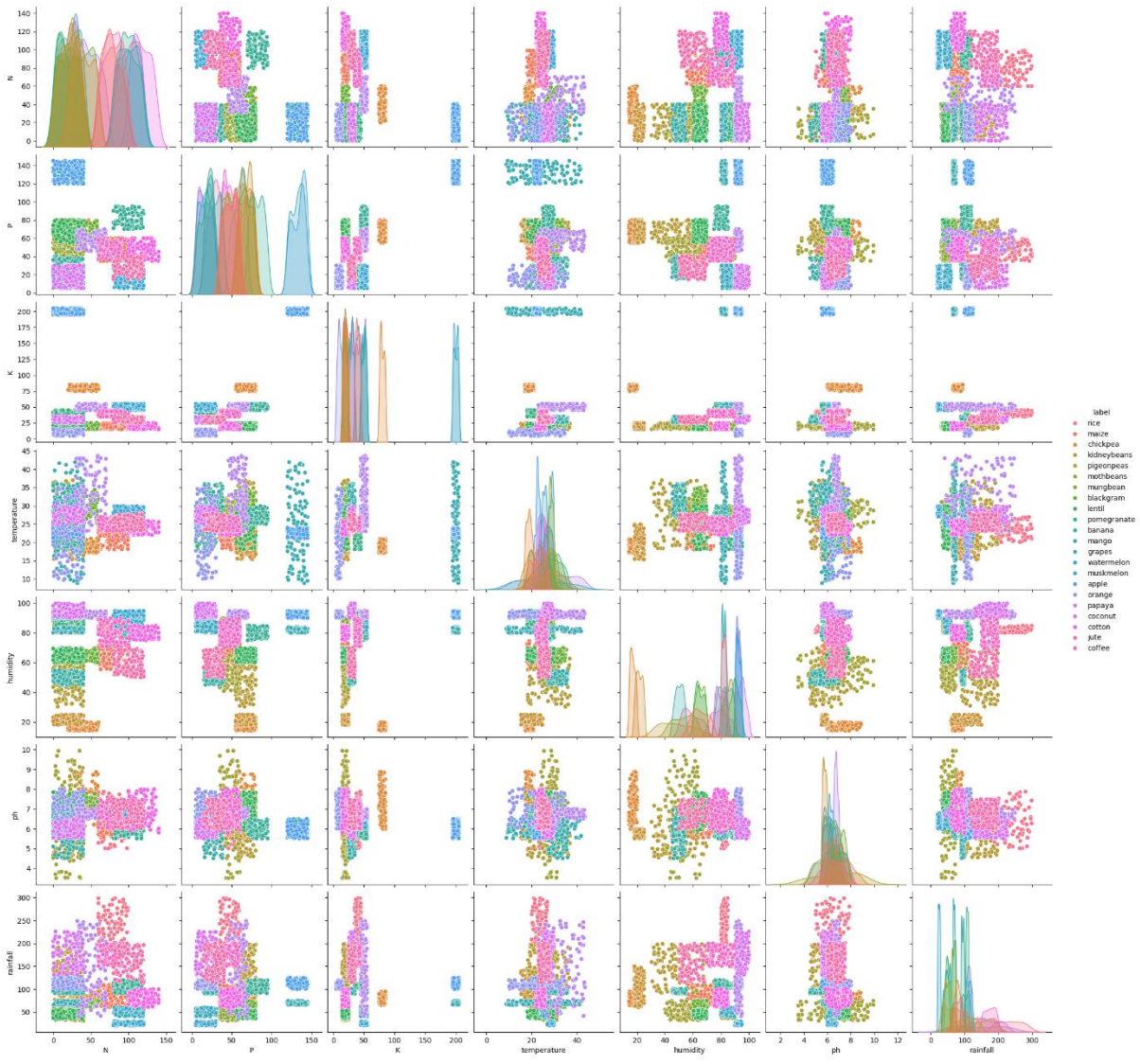

```python
#get correlations of each features in dataset
import pandas as pd
#If crop_df is not already a DataFrame,convert it
if isinstance(crop_df,np.ndarray):
    crop_df=pd.DataFrame(crop_df)
#Calculate the correlation matrix
corrmat=crop_df.select_dtypes(include=['number']).corr()
#Apply the background gradient styling
corrmat.style.background_gradient('coolwarm')
```

	N	P	K	temperature	humidity	ph	rainfall
N	1.000000	-0.231460	-0.140512	0.026504	0.190688	0.096683	0.059020
P	-0.231460	1.000000	0.736232	-0.127541	-0.118734	-0.138019	-0.063839
K	-0.140512	0.736232	1.000000	-0.160387	0.190859	-0.169503	-0.053461
temperature	0.026504	-0.127541	-0.160387	1.000000	0.205320	-0.017795	-0.030084
humidity	0.190688	-0.118734	0.190859	0.205320	1.000000	-0.008483	0.094423
ph	0.096683	-0.138019	-0.169503	-0.017795	-0.008483	1.000000	-0.109069
rainfall	0.059020	-0.063839	-0.053461	-0.030084	0.094423	-0.109069	1.000000

```
#Heatmap
sns.heatmap(corrmat, annot=True, cmap="RdYlGn")
plt.title("Correlation Matrix")
plt.show()
```




```
#Pairplot
sns.pairplot(crop_df, hue="label", height=3)
plt.show()
```



Splitting data into training and testing data

```
#X consist of independent variables
X= crop_df.drop(['label'], axis=1)
#Y consist of dependent variables
Y= pd.Categorical(crop_df.label)
```

```
X_train, X_test, Y_train, Y_test =  train_test_split(X,Y, test_size=0.2)
```

Model Building : Training the model in multiple Algorithms

```
#K-Nearest-Neighbors Classifier
knnclassifier=KNeighborsClassifier(n_neighbors=9)
knnclassifier.fit(X_train,y_train)
print("The accuracy of K Nearest Neighbors Classifier is",
      knnclassifier.score(X_train,y_train),knnclassifier.score(X_test,y_test))
knn=[knnclassifier.score(X_train,y_train),knnclassifier.score(X_test,y_test)]
```

```
The accuracy of K Nearest Neighbors Classifier is 0.9857954545454546 0.9704545454545455
```

```
#SVM Model
svm= SVC()
svm.fit(X_train, Y_train)
print("The accuracy of SVM is", svm.score(X_train, Y_train),svm.score(X_test, Y_test))
svm= [svm.score(X_train, Y_train),svm.score(X_test, Y_test)]
```

```
The accuracy of SVM is 0.977840909090909 0.975
```

```
#Decision Tree Model
dtclassifier= DecisionTreeClassifier(max_depth=7)
dtclassifier.fit(X_train, Y_train)
print("The accuracy of decision tree is", dtclassifier.score(X_train, Y_train),dtclassifier.score(X_test, Y_test))
dt= [dtclassifier.score(X_train, Y_train),dtclassifier.score(X_test, Y_test)]
```

```
The accuracy of decision tree is 0.9323863636363636 0.9318181818181818
```

```
#Random forest model
rfclassifier = RandomForestClassifier()
rfclassifier.fit(X_train, Y_train)
print("The accuracy of random forest classifier is", rfclassifier.score(X_train, Y_train), rfclassifier.score(X_test, Y_test))
rf= [rfclassifier.score(X_train, Y_train), rfclassifier.score(X_test, Y_test)]
```

```
The accuracy of random forest classifier is 1.0 0.9931818181818182
```

```
#evaluation metrics
#Knn
knnclassifier= KNeighborsClassifier()
knnclassifier.fit(X_train, Y_train)
y_pred = knnclassifier.predict(X_test)
print(classification_report(Y_test, y_pred))
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	23
banana	1.00	1.00	1.00	20
blackgram	0.91	1.00	0.95	21
chickpea	1.00	1.00	1.00	22
coconut	1.00	1.00	1.00	20
coffee	1.00	1.00	1.00	24
cotton	0.95	1.00	0.98	20
grapes	1.00	1.00	1.00	20
jute	0.74	0.94	0.83	18
kidneybeans	0.90	1.00	0.95	19
lentil	1.00	0.96	0.98	25
maize	1.00	0.90	0.95	20
mango	1.00	1.00	1.00	17
mothbeans	0.93	0.93	0.93	14
mungbean	1.00	1.00	1.00	20
muskmelon	1.00	1.00	1.00	18
orange	1.00	1.00	1.00	24
papaya	1.00	0.93	0.97	15
pigeonpeas	1.00	0.90	0.95	21
pomegranate	1.00	1.00	1.00	23
rice	0.93	0.74	0.82	19
watermelon	1.00	1.00	1.00	17
accuracy			0.97	440
macro avg	0.97	0.97	0.97	440
weighted avg	0.97	0.97	0.97	440

```
#Decision tree
dtclassifier= DecisionTreeClassifier()
dtclassifier.fit(X_train, Y_train)
y_pred = dtclassifier.predict(X_test)
print(classification_report(Y_test, y_pred))
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	23
banana	1.00	1.00	1.00	20
blackgram	0.84	1.00	0.91	21
chickpea	1.00	1.00	1.00	22
coconut	1.00	1.00	1.00	20
coffee	1.00	1.00	1.00	24
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	20
jute	0.89	0.94	0.92	18
kidneybeans	1.00	1.00	1.00	19
lentil	0.92	0.92	0.92	25
maize	1.00	1.00	1.00	20
mango	1.00	1.00	1.00	17
mothbeans	1.00	0.71	0.83	14
mungbean	1.00	1.00	1.00	20
muskmelon	1.00	1.00	1.00	18
orange	1.00	1.00	1.00	24
papaya	1.00	1.00	1.00	15
pigeonpeas	1.00	1.00	1.00	21
pomegranate	1.00	1.00	1.00	23
rice	0.94	0.89	0.92	19
watermelon	1.00	1.00	1.00	17
accuracy			0.98	440
macro avg	0.98	0.98	0.98	440
weighted avg	0.98	0.98	0.98	440

```
rfclassifier=RandomForestClassifier()
rfclassifier.fit(X_train,y_train)
y_pred=rfclassifier.predict(X_test)
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	23
banana	1.00	1.00	1.00	20
blackgram	1.00	1.00	1.00	21
chickpea	1.00	1.00	1.00	22
coconut	1.00	1.00	1.00	20
coffee	1.00	1.00	1.00	24
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	20
jute	0.90	1.00	0.95	18
kidneybeans	1.00	1.00	1.00	19
lentil	1.00	1.00	1.00	25
maize	1.00	1.00	1.00	20
mango	1.00	1.00	1.00	17
mothbeans	1.00	1.00	1.00	14
mungbean	1.00	1.00	1.00	20
muskmelon	1.00	1.00	1.00	18
orange	1.00	1.00	1.00	24
papaya	1.00	1.00	1.00	15
pigeonpeas	1.00	1.00	1.00	21
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.89	0.94	19
watermelon	1.00	1.00	1.00	17
accuracy			1.00	440
macro avg	1.00	1.00	1.00	440
weighted avg	1.00	1.00	1.00	440