

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

```
In [3]: data = pd.read_csv("Downloads/Crop_recommendationV2.csv")
```

```
In [5]: data
```

```
Out[5]:
```

	N	P	K	temperature	humidity	ph	rainfall	label	soil_moisture
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	29.446064
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	12.851183
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	29.363913
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	26.207732
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	28.236236
...	...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee	10.697757
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee	12.203830
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee	28.989176
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee	13.642305
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee	23.911728

2200 rows × 23 columns

```
In [7]: data.tail()
```

```
Out[7]:
```

	N	P	K	temperature	humidity	ph	rainfall	label	soil_moisture
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee	10.697757
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee	12.203830
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee	28.989176
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee	13.642305
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee	23.911728

5 rows × 23 columns

```
In [9]: data.head()
```

Out[9]:	N	P	K	temperature	humidity	ph	rainfall	label	soil_moisture	soil_1
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	29.446064	
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	12.851183	
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	29.363913	
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	26.207732	
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	28.236236	

5 rows × 23 columns

In [11]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 23 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
8   soil_moisture                         2200 non-null   float64
9   soil_type                             2200 non-null   int64
10  sunlight_exposure                     2200 non-null   float64
11  wind_speed                            2200 non-null   float64
12  co2_concentration                     2200 non-null   float64
13  organic_matter                        2200 non-null   float64
14  irrigation_frequency                  2200 non-null   int64
15  crop_density                          2200 non-null   float64
16  pest_pressure                         2200 non-null   float64
17  fertilizer_usage                      2200 non-null   float64
18  growth_stage                          2200 non-null   int64
19  urban_area_proximity                  2200 non-null   float64
20  water_source_type                     2200 non-null   int64
21  frost_risk                            2200 non-null   float64
22  water_usage_efficiency                 2200 non-null   float64
dtypes: float64(15), int64(7), object(1)
memory usage: 395.4+ KB
```

In [13]: `data.head(10)`

Out[13]:	N	P	K	temperature	humidity	ph	rainfall	label	soil_moisture	soil_t
<b>0</b>	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	29.446064	
<b>1</b>	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	12.851183	
<b>2</b>	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	29.363913	
<b>3</b>	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	26.207732	
<b>4</b>	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	28.236236	
<b>5</b>	69	37	42	23.058049	83.370118	7.073454	251.055000	rice	23.613115	
<b>6</b>	69	55	38	22.708838	82.639414	5.700806	271.324860	rice	15.333693	
<b>7</b>	94	53	40	20.277744	82.894086	5.718627	241.974195	rice	20.835640	
<b>8</b>	89	54	38	24.515881	83.535216	6.685346	230.446236	rice	26.640656	
<b>9</b>	68	58	38	23.223974	83.033227	6.336254	221.209196	rice	24.368853	

10 rows × 23 columns

```
In [15]: data.isnull().sum()
```

```
Out[15]: N          0
P          0
K          0
temperature  0
humidity     0
ph           0
rainfall     0
label        0
soil_moisture 0
soil_type    0
sunlight_exposure 0
wind_speed   0
co2_concentration 0
organic_matter 0
irrigation_frequency 0
crop_density 0
pest_pressure 0
fertilizer_usage 0
growth_stage 0
urban_area_proximity 0
water_source_type 0
frost_risk    0
water_usage_efficiency 0
dtype: int64
```

```
In [21]: X = data.drop("label", axis=1)
y = data["label"]
```

```
In [29]: from sklearn.preprocessing import LabelEncoder
```

```
In [31]: encoder = LabelEncoder()
y_encoded = encoder.fit_transform(y)
```

```
In [33]: X_train, X_test, y_train, y_test = train_test_split(
X, y_encoded, test_size=0.2, random_state=42
)
```

```
In [35]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [39]: from sklearn.ensemble import RandomForestClassifier
```

```
In [41]: model = RandomForestClassifier(n_estimators=200, random_state=42)
model.fit(X_train, y_train)
```

```
Out[41]: ▼ RandomForestClassifier
RandomForestClassifier(n_estimators=200, random_state=42)
```

```
In [44]: from sklearn.metrics import accuracy_score, classification_report
```

```
In [46]: y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 0.9886363636363636

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	21
2	1.00	1.00	1.00	20
3	1.00	1.00	1.00	26
4	1.00	1.00	1.00	27
5	1.00	1.00	1.00	17
6	1.00	1.00	1.00	17
7	1.00	1.00	1.00	14
8	0.92	0.96	0.94	23
9	1.00	1.00	1.00	20
10	0.85	1.00	0.92	11
11	1.00	1.00	1.00	21
12	1.00	1.00	1.00	19
13	1.00	0.92	0.96	24
14	1.00	1.00	1.00	19
15	1.00	1.00	1.00	17
16	1.00	1.00	1.00	14
17	1.00	1.00	1.00	23
18	1.00	1.00	1.00	23
19	1.00	1.00	1.00	23
20	0.94	0.89	0.92	19
21	1.00	1.00	1.00	19
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

```
In [ ]:
```