

In [4]:

```
#importing the required libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
```

In [5]:

```
data=pd.read_csv('cpdata.csv')
print(data.head(10))
```

	temperature	humidity	ph	rainfall	label
0	20.879744	82.002744	6.502985	202.935536	rice
1	21.770462	80.319644	7.038096	226.655537	rice
2	23.004459	82.320763	7.840207	263.964248	rice
3	26.491096	80.158363	6.980401	242.864034	rice
4	20.130175	81.604873	7.628473	262.717340	rice
5	23.058049	83.370118	7.073454	251.055000	rice
6	22.708838	82.639414	5.700806	271.324860	rice
7	20.277744	82.894086	5.718627	241.974195	rice
8	24.515881	83.535216	6.685346	230.446236	rice
9	23.223974	83.033227	6.336254	221.209196	rice

In [6]:

```
# Get the count of rows and columns
data.shape
```

Out[6]:

(3100, 5)

In [7]:

```
data.dtypes
```

Out[7]:

```
temperature    float64
humidity       float64
ph             float64
rainfall       float64
label          object
dtype: object
```

In [8]:

```
data.describe()
```

Out[8]:

	temperature	humidity	ph	rainfall
count	3100.000000	3100.000000	3100.000000	3100.000000
mean	27.108466	66.005312	6.368913	110.213031
std	7.566308	24.007713	0.809477	64.048562
min	8.825675	10.034048	3.504752	20.211267
25%	22.810495	55.244920	5.895343	64.909095
50%	26.102848	68.980529	6.342518	97.057093
75%	29.365644	84.446524	6.841616	141.210784
max	54.986760	99.981876	9.935091	397.315380

In [9]:

```
data['label'].unique()
```

Out[9]:

```
array(['rice', 'wheat', 'Mung Bean', 'Tea', 'millet', 'maize', 'Lentil',  
      'Jute', 'Coffee', 'Cotton', 'Ground Nut', 'Peas', 'Rubber',  
      'Sugarcane', 'Tobacco', 'Kidney Beans', 'Moth Beans', 'Coconut',  
      'Black gram', 'Adzuki Beans', 'Pigeon Peas', 'Chickpea', 'banana',  
      'grapes', 'apple', 'mango', 'muskmelon', 'orange', 'papaya',  
      'pomegranate', 'watermelon'], dtype=object)
```

In [10]:

```
cleanup_nums = {"label":      {'rice':0, 'wheat':1, 'Mung Bean':2, 'Tea':3, 'millet':4, 'maize':5, 'Jute':7, 'Coffee':8, 'Cotton':9, 'Ground Nut':10, 'Peas':11, 'Rubber':12, 'Sugarcane':13, 'Tobacco':14, 'Kidney Beans':15, 'Moth Beans':16, 'Coconut':17, 'Black gram':18, 'Adzuki Beans':19, 'Pigeon Peas':20, 'Chickpea':21, 'banana':22, 'grapes':23, 'apple':24, 'mango':25, 'muskmelon':26, 'orange':27, 'papaya':28, 'pomegranate':29, 'watermelon':30}}
```

```
data.replace(cleanup_nums, inplace=True)
```

```
data.head(10)
```

Out[10]:

	temperature	humidity	ph	rainfall	label
0	20.879744	82.002744	6.502985	202.935536	0
1	21.770462	80.319644	7.038096	226.655537	0
2	23.004459	82.320763	7.840207	263.964248	0
3	26.491096	80.158363	6.980401	242.864034	0
4	20.130175	81.604873	7.628473	262.717340	0
5	23.058049	83.370118	7.073454	251.055000	0
6	22.708838	82.639414	5.700806	271.324860	0
7	20.277744	82.894086	5.718627	241.974195	0
8	24.515881	83.535216	6.685346	230.446236	0
9	23.223974	83.033227	6.336254	221.209196	0

In [11]:

```
# Split the data into independent 'X' and dependent 'y'.
```

```
X=data.iloc[:, 0:4].values
```

```
y=data.iloc[:, 4].values
```

In [12]:

```
#Dividing the data into training and test set
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3)
```

In [196]:

```
from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
```

```
X_test = sc.transform(X_test)
```

In [13]:

```
#Importing Logistic Regression
```

```
from sklearn.linear_model import LogisticRegression
```

```
clf=LogisticRegression()
```

In [14]:

```
#Fitting the classifier into training set
clf.fit(X_train,y_train)
pred=clf.predict(X_test)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model_logistic.p
y:762: 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> (<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 (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
n_iter_i = _check_optimize_result(

In [15]:

```
from sklearn.metrics import accuracy_score
# Finding the accuracy of the model
a=accuracy_score(y_test,pred)
print("The accuracy of this model is: ", a*100)
```

The accuracy of this model is: 58.602150537634415

In [17]:

```
import sklearn.externals
import joblib
joblib.dump(clf,"crop_prediction.pkl")
```

Out[17]:

['crop_prediction.pkl']

In [16]:

```
ran_data = [20,82,6,202]
ran_data_arr = np.array(ran_data)
ran_data_num = ran_data_arr.reshape(1,-1)
pred_single_row = clf.predict(ran_data_num)
print(pred_single_row)
```

[0]

In []:

