In [4]:
```
# for manipulation
import numpy as np
import pandas as pd


# for data visulization
import matplotlib.pyplot as plt
import seaborn as sns

#for data interactivity
from ipywidgets import interact
```

In [5]:
```
#read the dataset
data=pd.read_csv('data.csv')
```

In [6]:
```
# check the shape of data set
print("shape of the dataset :",data.shape)
```

```
shape of the dataset : (2200, 8)
```

In [7]:
```
#check the head of the data set
data.head
```

Out[7]:
```
<bound method NDFrame.head of         N   P   K  temperature   humidity        ph    rai
nfall   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 x 8 columns]>
```

In [8]:
```
#check the  missing values
data.isnull().sum()
```

Out[8]:
```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

In [9]:
```
#check the crops present in the dataset
data['label'].value_counts()
```

Out[9]:
```
kidneybeans    100
papaya         100
banana         100
mango          100
rice           100
blackgram      100
```

```
chickpea       100
orange         100
grapes         100
cotton         100
apple          100
muskmelon      100
mothbeans      100
pomegranate    100
mungbean       100
jute           100
maize          100
coconut        100
watermelon     100
lentil         100
pigeonpeas     100
coffee         100
Name: label, dtype: int64
```

In [10]:
```
# The Average condition of all crops

print("Average ratio of the nitrogen in the soil: {0:2f}".format(data['N'].mean()))
print("Average ratio of the Phosphorous in the soil: {0:2f}".format(data['P'].mean()))
print("Average ratio of the Potassium in the soil: {0:2f}".format(data['K'].mean()))
print("Average Temperature in celsius: {0:2f}".format(data['temperature'].mean()))
print("Average Relative Humidity in the soil: {0:2f}".format(data['humidity'].mean()))
print("Average PH value in the soil: {0:2f}".format(data['ph'].mean()))
print("Average rainfall in mm: {0:2f}".format(data['rainfall'].mean()))
```

```
Average ratio of the nitrogen in the soil: 50.551818
Average ratio of the Phosphorous in the soil: 53.362727
Average ratio of the Potassium in the soil: 48.149091
Average Temperature in celsius: 25.616244
Average Relative Humidity in the soil: 71.481779
Average PH value in the soil: 6.469480
Average rainfall in mm: 103.463655
```

In [11]:
```
#check the summary statistics for each of the crops
@interact
def summary(crops=list(data['label'].value_counts().index)):
    x=data[data['label']==crops]

    print("------------------------------------------------")
    print("Statics for Nitrogen")
    print("Minimum Nitrogen required :",x['N'].min())
    print("Average Nitrogen required :",x['N'].mean())
    print("Maximum Notrogen required :",x['N'].max())

    print("------------------------------------------------")
    print("Statics for Phosphorous")
    print("Minimum phosphorous required :",x['P'].min())
    print("Average phosphorous required :",x['P'].mean())
    print("Maximum phosphorous required :",x['P'].max())

    print("------------------------------------------------")
    print("Statics for Potassium")
    print("Minimum Potassium required :{0:2f}",x['K'].min())
    print("Average Potassium required :{0:2f}",x['K'].mean())
    print("Maximum Potassium required :{0:2f}",x['K'].max())

    print("------------------------------------------------")
    print("Statics for Temperature")
    print("Minimum Temperature required :{0:2f}",x['temperature'].min())
    print("Average Temperature required :{0:2f}",x['temperature'].mean())
```

```
    print("Maximum Temperature required :{0:2f}",x['temperature'].max())

    print("-----------------------------------------------")
    print("Statics for Humidity")
    print("Minimum Humidity required :{0:2f}",x['humidity'].min())
    print("Average Humidity required :{0:2f}",x['humidity'].mean())
    print("Maximum Humidity required :{0:2f}",x['humidity'].max())

    print("-----------------------------------------------")
    print("Statics for PH")
    print("Minimum PH required :{0:2f}",x['ph'].min())
    print("Average PH required :{0:2f}",x['ph'].mean())
    print("Maximum PH required :{0:2f}",x['ph'].max())

    print("-----------------------------------------------")
    print("Statics for Rainfall")
    print("Minimum Rainfall required :{0:2f}",x['rainfall'].min())
    print("Average Rainfall required :{0:2f}",x['rainfall'].mean())
    print("Maximum Rainfall required :{0:2f}",x['rainfall'].max())
```

In [12]:

```
#compare the average requirement for each crops with average condition
@interact
def compare(conditions =['N','P','K','temperature','ph','humidity','rainfall']):
    print("Average value for",conditions,"is{0:2f}".format(data[conditions].mean()))
    print("--------------------------------------------------------------------")
    print("Rice :{0:2f}".format(data[(data['label']=='rice')][conditions].mean()))
    print("Black Grams :{0:2f}".format(data[(data['label']=='blackgrams')][conditions].
    print("Banana :{0:2f}".format(data[(data['label']=='banana')][conditions].mean()))
    print("Jute :{0:2f}".format(data[(data['label']=='jute')][conditions].mean()))
    print("Coconut :{0:2f}".format(data[(data['label']=='coconut')][conditions].mean())
    print("Apple :{0:2f}".format(data[(data['label']=='apple')][conditions].mean()))
    print("Papaya :{0:2f}".format(data[(data['label']=='papaye')][conditions].mean()))
    print("Muskmelon :{0:2f}".format(data[(data['label']=='muskmelon')][conditions].mea
    print("Grapes :{0:2f}".format(data[(data['label']=='grapes')][conditions].mean()))
    print("Watermelon :{0:2f}".format(data[(data['label']=='watermelon')][conditions].m
    print("Kidney Beans :{0:2f}".format(data[(data['label']=='kidneybeans')][conditions
    print("Mung Beans :{0:2f}".format(data[(data['label']=='mungbeans')][conditions].me
    print("Oranges :{0:2f}".format(data[(data['label']=='oranges')][conditions].mean())
    print("Chick peas :{0:2f}".format(data[(data['label']=='chickpeas')][conditions].me
    print("Lentil :{0:2f}".format(data[(data['label']=='lentil')][conditions].mean()))
    print("Cotton :{0:2f}".format(data[(data['label']=='cotton')][conditions].mean()))
    print("Maize :{0:2f}".format(data[(data['label']=='maize')][conditions].mean()))
    print("Moth Beans :{0:2f}".format(data[(data['label']=='moth beans')][conditions].m
    print("Pigeon Peas :{0:2f}".format(data[(data['label']=='pigeonpeas')][conditions].
    print("Mango :{0:2f}".format(data[(data['label']=='mango')][conditions].mean()))
    print("Pomegranate :{0:2f}".format(data[(data['label']=='pomegranate')][conditions]
    print("Coffee :{0:2f}".format(data[(data['label']=='coffee')][conditions].mean()))
```

In [13]:

```
#Makes this function more intuitive
@interact
def compare(conditions =['N','P','K','temperature','ph','humidity','rainfall']):
    print("crops which require greater than average",conditions,'\n')
    print(data[data[conditions]>data[conditions].mean()]['label'].unique())
    print("----------------------------------------------------------")
    print("crops which require less than average",'\n')
    print(data[data[conditions]<=data[conditions].mean()]['label'].unique())
```

In [32]:
```
#Some interesting facts

print("--------------------------------")
print("crops which require very high ratio of Nitrogen content in soil:",data[data['N']
print("crops which require very high ratio of Phosphorous content in soil:",data[data['
print("crops which require very high ratio of Potassium content in soil:",data[data['K'
print("crops which require very high Rainfall:",data[data['rainfall']>200]['label'].uni
print("crops which require very low Temperature:",data[data['temperature']<10]['label']
print("crops which require very high Temperature:",data[data['temperature']>40]['label'
print("crops which require very Low Humidity:",data[data['humidity']<20]['label'].uniqu
print("crops which require very Low pH:",data[data['ph']<4]['label'].unique())
print("crops which require very high pH:",data[data['ph']>9]['label'].unique())
```

```
--------------------------------
crops which require very high ratio of Nitrogen content in soil: ['cotton']
crops which require very high ratio of Phosphorous content in soil: ['grapes' 'apple']
crops which require very high ratio of Potassium content in soil: ['grapes' 'apple']
crops which require very high Rainfall: ['rice' 'papaya' 'coconut']
crops which require very low Temperature: ['grapes']
crops which require very high Temperature: ['grapes' 'papaya']
crops which require very Low Humidity: ['chickpea' 'kidneybeans']
crops which require very Low pH: ['mothbeans']
crops which require very high pH: ['mothbeans']
```

In [38]:
```
#crops which Grown in summer season,winter season and Rainy season

print("Summer crops")
print(data[(data['temperature']>30)&(data['humidity']>50)]['label'].unique())
print("-----------------------------------------------")
print("winter crops")
print(data[(data['temperature']<20)&(data['humidity']>30)]['label'].unique())
print("-----------------------------------------------")
print("Rainy crops")
print(data[(data['rainfall']>200)&(data['humidity']>30)]['label'].unique())
```

```
Summer crops
['pigeonpeas' 'mothbeans' 'blackgram' 'mango' 'grapes' 'orange' 'papaya']
-----------------------------------------------
winter crops
['maize' 'pigeonpeas' 'lentil' 'pomegranate' 'grapes' 'orange']
-----------------------------------------------
Rainy crops
['rice' 'papaya' 'coconut']
```

In [14]:
```
#split of data set for predective modelling

y=data['label']
x=data.drop(['label'],axis=1)

print("shape of x:",x.shape)
print("shape of y:",y.shape)
```

```
shape of x: (2200, 7)
shape of y: (2200,)
```

In [18]:
```
#creating traning and testing sets for validation of result
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,random_state=0)

print("The shape of x train:",x_train.shape)
```

```
print("The shape of x test:",x_test.shape)
print("The shape of y train:",x_train.shape)
print("The shape of y test:",x_test.shape)
```

```
The shape of x train: (1760, 7)
The shape of x test: (440, 7)
The shape of y train: (1760, 7)
The shape of y test: (440, 7)
```

In [19]:
```
#Lets create a predective model

from sklearn.linear_model import logisticRegression

model=logisticRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-19-f3148d4f6f65> in <module>
      1 #lets create a predective model
      2
----> 3 from sklearn.linear_model import logisticRegression
      4
      5 model=logisticRegression()

ImportError: cannot import name 'logisticRegression' from 'sklearn.linear_model' (C:\pyt
hon anaconda mohit\lib\site-packages\sklearn\linear_model\__init__.py)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: