ADA Task(232-25-034).ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

MD.AHSAN HABIB(232-25-034)

ADA TASK- FALL 2023

IMPORTING ALL NECESSARY LIBRARIES

```
[136] import numpy as np
      import pandas as pd
      from google.colab import files
      from sklearn.cluster import KMeans
      import matplotlib.pyplot as plt
      from sklearn.cluster import DBSCAN
      from sklearn.preprocessing import StandardScaler
```

IMPORTING GIVEN DATASET

```
[137] from google.colab import files
      files.upload()
```

Choose Files NPK Dataset.csv

- **NPK Dataset.csv**(text/csv) - 1493 bytes, last modified: 11/7/2023 - 100% done

```
Saving NPK Dataset.csv to NPK Dataset (3).csv
{'NPK Dataset (3).csv': b'Date,Time,Input Organic Fertilizer (gm),Nitrogen,Phosporus,Potasium,Temperature,Moisture,Location,Organic Fertilizer\r\n29.10.2023,11:05 AM,100,48 mg/kg,119 mg/kg,107
mg/kg,26,859,"Ashulia,Savar,Dhaka",\r\n29.10.2023,12:45 PM,,50 mg/kg,124 mg/kg,113 mg/kg,27,863,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:20 AM,,53 mg/kg,120 mg/kg,118 mg/kg,29,853,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:25 AM,,54
mg/kg,122 mg/kg,120 mg/kg,29,873,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:30 AM,,54 mg/kg,122 mg/kg,120 mg/kg,29,852,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:45 AM,,56 mg/kg,123 mg/kg,121
mg/kg,29,870,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:46 AM,,56 mg/kg,123 mg/kg,121 mg/kg,29,861,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:47 AM,,56 mg/kg,123 mg/kg,121 mg/kg,29,861,"Ashulia,Savar,Dhaka",\r\n30.10.2023,10:50 AM,,56
mg/kg,123 mg/kg,121 mg/kg,29,872,"Ashulia,Savar,Dhaka",\r\n30.10.2023,11:10 AM,,57 mg/kg,123 mg/kg,121 mg/kg,29,875,"Ashulia,Savar,Dhaka",\r\n31.10.2023,12:46 PM,,47 mg/kg,117 mg/kg,107
mg/kg,28,857,"Ashulia,Savar,Dhaka",\r\n31.10.2023,12:48 PM,,47 mg/kg,117 mg/kg,107 mg/kg,28,867,"Ashulia,Savar,Dhaka",\r\n31.10.2023,12:50 PM,,47 mg/kg,117 mg/kg,107 mg/kg,28,857,"Ashulia,Savar,Dhaka",\r\n31.10.2023,12:51 PM,,47
mg/kg,117 mg/kg,107 mg/kg,28,855,"Ashulia,Savar,Dhaka",\r\n01.11.2023,2:15 PM,,53 mg/kg,120 mg/kg,118 mg/kg,26,857,"Ashulia,Savar,Dhaka",\r\n01.11.2023,2:17 PM,,53 mg/kg,120 mg/kg,118
mg/kg,26,860,"Ashulia,Savar,Dhaka",\r\n01.11.2023,2:20 PM,,47 mg/kg,120 mg/kg,118 mg/kg,26,873,"Ashulia,Savar,Dhaka",'}
```

READING DATASET

```
[138] df = pd.read_csv('NPK Dataset.csv')
      df.head()
```

	Date	Time	Input Organic Fertilizer (gm)	Nitrogen	Phosporus	Potasium	Temperature	Moisture	Location	Organic Fertilizer
0	29.10.2023	11:05 AM	100.0	48 mg/kg	119 mg/kg	107 mg/kg	26	859	Ashulia,Savar,Dhaka	NaN
1	29.10.2023	12:45 PM	NaN	50 mg/kg	124 mg/kg	113 mg/kg	27	863	Ashulia,Savar,Dhaka	NaN
2	30.10.2023	10:20 AM	NaN	53 mg/kg	120 mg/kg	118 mg/kg	29	853	Ashulia,Savar,Dhaka	NaN
3	30.10.2023	10:25 AM	NaN	54 mg/kg	122 mg/kg	120 mg/kg	29	873	Ashulia,Savar,Dhaka	NaN
4	30.10.2023	10:30 AM	NaN	54 mg/kg	122 mg/kg	120 mg/kg	29	852	Ashulia,Savar,Dhaka	NaN

```
[139] print("Dataset Shape=", df.shape)
      print("\nDataset Overview with all possible values")
      print(df.columns)
```

```
Dataset Shape= (17, 10)

Dataset Overview with all possible values
Index(['Date', 'Time', 'Input Organic Fertilizer (gm)', 'Nitrogen',
       'Phosporus', 'Potasium', 'Temperature', 'Moisture', 'Location',
       'Organic Fertilizer'],
      dtype='object')
```

PREPROCESSING: Formatting Date and Time column appropriately

```
[140] df['Date_Time'] = df['Date']+ ' '+df['Time']
      df['Date_Time']  = pd.to_datetime(df['Date_Time'], format="%d.%m.%Y %I:%M %p", dayfirst=True)

      df['Date'] = df['Date_Time'].dt.date
      df['Time'] = df['Date_Time'].dt.time

      df = df.drop(columns=['Date_Time'])
```

PREPROCESSING: Dropping Unnecessary Columns

```
[141] df = df.drop(columns=['Organic Fertilizer'])
      df = df.drop(columns=['Input Organic Fertilizer (gm)'])
      print(df.columns)
```

```
Index(['Date', 'Time', 'Nitrogen', 'Phosporus', 'Potasium', 'Temperature',
       'Moisture', 'Location'],
      dtype='object')
```

PREPROCESSING: Removing string units from numeric data

```
[142] df['Nitrogen'] = df['Nitrogen'].str.replace('mg/kg', '').astype(int)
      df['Phosporus'] = df['Phosporus'].str.replace('mg/kg', '').astype(int)
      df['Potasium'] = df['Potasium'].str.replace('mg/kg', '').astype(int)

      df.rename(columns={
          'Nitrogen' : 'Nitrogen(mg/kg)',
          'Phosporus' : 'Phosporus(mg/kg)',
          'Potasium' : 'Potasium(mg/kg)'
      }, inplace=True)

      print(df.head(3))
```

```
         Date      Time  Nitrogen(mg/kg)  Phosporus(mg/kg)  Potasium(mg/kg)  \
0  2023-10-29  11:05:00               48               119              107
1  2023-10-29  12:45:00               50               124              113
2  2023-10-30  10:20:00               53               120              118

   Temperature  Moisture            Location
0           26       859  Ashulia,Savar,Dhaka
1           27       863  Ashulia,Savar,Dhaka
2           29       853  Ashulia,Savar,Dhaka
```

ADDING NEW DUMMY LOCATION: We made 20 instance of dataset, assigned new locations to each

```
[143] df0=df.copy()
      df1=df.copy()
      df2=df.copy()
      df3=df.copy()
      df4=df.copy()
```

```
df5=df.copy()
df6=df.copy()
df7=df.copy()
df8=df.copy()
df9=df.copy()
df10=df.copy()
df11=df.copy()
df12=df.copy()
df13=df.copy()
df14=df.copy()
df15=df.copy()
df16=df.copy()
df17=df.copy()
df18=df.copy()
df19=df.copy()

df0['Location'] = "Tongi, Gazipur "
df1['Location'] = "Ashulia, Dhaka "
df2['Location'] = "Mirpur, Dhaka"
df3['Location'] = "Konabari, Gazipur"
df4['Location'] = "House Building, Uttara"
df5['Location'] = "Banani , Dhaka"
df6['Location'] = "Abdullahpur , Dhaka"
df7['Location'] = "Mirpur-10 , Dhaka"
df8['Location'] = "Board Bazar , Gazipur"
df9['Location'] = "Savar , Dhaka"
df10['Location'] = "Dhanmondi , Dhaka"
df11['Location'] = "Shonir Akhra , Dhaka"
df12['Location'] = "Mohammadepur , Dhaka"
df13['Location'] = "Chow Rasta , Gazipur"
df14['Location'] = "Dhamrai , Dhaka"
df15['Location'] = "Gabtoli , Dhaka"
df16['Location'] = "Uttara Diyabari , Dhaka"
df17['Location'] = "Aftabnogor , Dhaka"
df18['Location'] = "Bosila, Dhaka"
df19['Location'] = "Mirpur-12 , Dhaka"
```

ADDING NEW DUMMY DATA: Using random function we made dummy data

```
[144] def dummy (c,d,r,w):
          modified_values = []
          if r == 0:
              r = [1,2]
          if w == 0:
              w = [0.7, 0.3]
          random_numbers = np.random.choice(r,size=len(d))
          operations = np.random.choice([1,-1],size=len(d),p=w)
          iv = d[c].iloc[0]
          modified_values = [iv]


          for i in range(1,len(d)):
              previous_value = modified_values[i-1]
              random_number = random_numbers[i]
              operation = operations[i]
              if previous_value <= iv:
                  operation =1
              new_value = previous_value + (random_number * operation)
              modified_values.append(new_value)
          d[c]= modified_values



      dummy('Nitrogen(mg/kg)', df0,0,0)
      dummy('Phosporus(mg/kg)', df0,0,0)
      dummy('Potasium(mg/kg)', df0,0,0)
      dummy('Temperature', df0, [1],[0.5,0.5])
      dummy('Moisture', df0, [1,2,3],[0.5,0.5])

      dummy('Nitrogen(mg/kg)', df1,0,0)
      dummy('Phosporus(mg/kg)', df1,0,0)
      dummy('Potasium(mg/kg)', df1,0,0)
      dummy('Temperature', df1, [1],[0.5,0.5])
      dummy('Moisture', df1, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df2,0,0)
      dummy('Phosporus(mg/kg)', df2,0,0)
      dummy('Potasium(mg/kg)', df2,0,0)
      dummy('Temperature', df2, [1],[0.5,0.5])
      dummy('Moisture', df2, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df3,0,0)
      dummy('Phosporus(mg/kg)', df3,0,0)
      dummy('Potasium(mg/kg)', df3,0,0)
      dummy('Temperature', df3, [1],[0.5,0.5])
      dummy('Moisture', df3, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df4,0,0)
      dummy('Phosporus(mg/kg)', df4,0,0)
      dummy('Potasium(mg/kg)', df4,0,0)
      dummy('Temperature', df4, [1],[0.5,0.5])
      dummy('Moisture', df4, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df5,0,0)
      dummy('Phosporus(mg/kg)', df5,0,0)
      dummy('Potasium(mg/kg)', df5,0,0)
      dummy('Temperature', df5, [1],[0.5,0.5])
      dummy('Moisture', df5, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df6,0,0)
      dummy('Phosporus(mg/kg)', df6,0,0)
      dummy('Potasium(mg/kg)', df6,0,0)
      dummy('Temperature', df6, [1],[0.5,0.5])
      dummy('Moisture', df6, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df7,0,0)
      dummy('Phosporus(mg/kg)', df7,0,0)
      dummy('Potasium(mg/kg)', df7,0,0)
      dummy('Temperature', df7, [1],[0.5,0.5])
      dummy('Moisture', df7, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df8,0,0)
      dummy('Phosporus(mg/kg)', df8,0,0)
      dummy('Potasium(mg/kg)', df8,0,0)
      dummy('Temperature', df8, [1],[0.5,0.5])
      dummy('Moisture', df8, [1,2,3],[0.5,0.5])


      dummy('Nitrogen(mg/kg)', df9,0,0)
      dummy('Phosporus(mg/kg)', df9,0,0)
      dummy('Potasium(mg/kg)', df9,0,0)
      dummy('Temperature', df9, [1],[0.5,0.5])
      dummy('Moisture', df9, [1,2,3],[0.5,0.5])
```

```
dummy('Nitrogen(mg/kg)', df10,0,0)
dummy('Phosporus(mg/kg)', df10,0,0)
dummy('Potasium(mg/kg)', df10,0,0)
dummy('Temperature', df10, [1],[0.5,0.5])
dummy('Moisture', df10, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df11,0,0)
dummy('Phosporus(mg/kg)', df11,0,0)
dummy('Potasium(mg/kg)', df11,0,0)
dummy('Temperature', df11, [1],[0.5,0.5])
dummy('Moisture', df11, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df12,0,0)
dummy('Phosporus(mg/kg)', df12,0,0)
dummy('Potasium(mg/kg)', df12,0,0)
dummy('Temperature', df12, [1],[0.5,0.5])
dummy('Moisture', df12, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df13,0,0)
dummy('Phosporus(mg/kg)', df13,0,0)
dummy('Potasium(mg/kg)', df13,0,0)
dummy('Temperature', df13, [1],[0.5,0.5])
dummy('Moisture', df13, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df14,0,0)
dummy('Phosporus(mg/kg)' ,df14,0,0)
dummy('Potasium(mg/kg)', df14,0,0)
dummy('Temperature', df14, [1],[0.5,0.5])
dummy('Moisture', df14, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df15,0,0)
dummy('Phosporus(mg/kg)', df15,0,0)
dummy('Potasium(mg/kg)', df15,0,0)
dummy('Temperature', df15, [1],[0.5,0.5])
dummy('Moisture', df15, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df16,0,0)
dummy('Phosporus(mg/kg)', df16,0,0)
dummy('Potasium(mg/kg)', df16,0,0)
dummy('Temperature', df16, [1],[0.5,0.5])
dummy('Moisture', df16, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df17,0,0)
dummy('Phosporus(mg/kg)', df17,0,0)
dummy('Potasium(mg/kg)', df17,0,0)
dummy('Temperature', df17, [1],[0.5,0.5])
dummy('Moisture' ,df17, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df18,0,0)
dummy('Phosporus(mg/kg)' ,df18,0,0)
dummy('Potasium(mg/kg)', df18,0,0)
dummy('Temperature', df18, [1],[0.5,0.5])
dummy('Moisture', df18, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)' ,df18,0,0)
dummy('Phosporus(mg/kg)', df18,0,0)
dummy('Potasium(mg/kg)', df18,0,0)
dummy('Temperature', df18, [1],[0.5,0.5])
dummy('Moisture' ,df18, [1,2,3],[0.5,0.5])


dummy('Nitrogen(mg/kg)', df19,0,0)
dummy('Phosporus(mg/kg)', df19,0,0)
dummy('Potasium(mg/kg)', df19,0,0)
dummy('Temperature', df19, [1],[0.5,0.5])
dummy('Moisture', df19, [1,2,3],[0.5,0.5])


dataframes = [df,df0, df1, df2, df3, df4, df5, df6, df7, df8, df9, df10, df11, df12, df13, df14, df15, df16, df17, df18, df19]
combined_df = pd.concat(dataframes, ignore_index=True)
```

DUMMY DATA SAVED & MERGED:

```
[145] combined_df.to_csv('agriculture.csv')
```

Downloading the new dataset to our local machine

```
[146] from google.colab import files
      files.download('agriculture.csv')
```

LOADING NEW DATASET

```
[147] from google.colab import files
      files.upload()
```


```
Dhaka"\n173,2023-10-30,10:25:00,53,120,109,27,860,"Savar , Dhaka"\n174,2023-10-30,10:30:00,55,121,110,26,859,"Savar , Dhaka"\n175,2023-10-30,10:45:00,56,120,109,27,860,"Savar , Dhaka"\n176,2023-10-
30,10:46:00,57,122,108,28,861,"Savar , Dhaka"\n177,2023-10-30,10:47:00,59,123,107,29,860,"Savar , Dhaka"\n178,2023-10-30,10:50:00,61,124,108,30,857,"Savar , Dhaka"\n179,2023-10-30,11:10:00,59,123,109,29,858,"Savar ,
Dhaka"\n180,2023-10-31,12:46:00,61,125,110,28,861,"Savar , Dhaka"\n181,2023-10-31,12:48:00,60,127,112,27,862,"Savar , Dhaka"\n182,2023-10-31,12:50:00,59,125,114,28,859,"Savar , Dhaka"\n183,2023-10-
31,12:51:00,61,124,116,29,861,"Savar , Dhaka"\n184,2023-11-01,14:15:00,63,125,118,28,864,"Savar , Dhaka"\n185,2023-11-01,14:17:00,65,123,119,29,861,"Savar , Dhaka"\n186,2023-11-01,14:20:00,66,122,118,28,864,"Savar ,
Dhaka"\n187,2023-10-29,11:05:00,48,119,107,26,859,"Dhanmondi , Dhaka"\n188,2023-10-29,12:45:00,50,121,108,27,861,"Dhanmondi , Dhaka"\n189,2023-10-30,10:20:00,52,122,110,26,864,"Dhanmondi , Dhaka"\n190,2023-10-
30,10:25:00,50,121,111,27,861,"Dhanmondi , Dhaka"\n191,2023-10-30,10:30:00,52,123,110,26,864,"Dhanmondi , Dhaka"\n192,2023-10-30,10:45:00,54,124,112,27,862,"Dhanmondi , Dhaka"\n193,2023-10-30,10:46:00,55,126,110,26,860,"Dhanmondi ,
Dhaka"\n194,2023-10-30,10:47:00,54,124,112,27,859,"Dhanmondi , Dhaka"\n195,2023-10-30,10:50:00,55,123,114,26,861,"Dhanmondi , Dhaka"\n196,2023-10-30,11:10:00,56,121,115,27,859,"Dhanmondi , Dhaka"\n197,2023-10-
31,12:46:00,54,122,116,26,862,"Dhanmondi , Dhaka"\n198,2023-10-31,12:48:00,55,123,118,27,861,"Dhanmondi , Dhaka"\n199,2023-10-31,12:50:00,53,124,119,26,864,"Dhanmondi , Dhaka"\n200,2023-10-31,12:51:00,54,125,121,27,867,"Dhanmondi ,
Dhaka"\n201,2023-11-01,14:15:00,55,127,119,26,866,"Dhanmondi , Dhaka"\n202,2023-11-01,14:17:00,56,129,121,27,868,"Dhanmondi , Dhaka"\n203,2023-11-01,14:20:00,54,131,119,28,870,"Dhanmondi , Dhaka"\n204,2023-10-
29,11:05:00,48,119,107,26,859,"Shonir Akhra , Dhaka"\n205,2023-10-29,12:45:00,49,121,108,27,860,"Shonir Akhra , Dhaka"\n206,2023-10-30,10:20:00,51,123,106,28,861,"Shonir Akhra , Dhaka"\n207,2023-10-
30,10:25:00,53,124,108,27,858,"Shonir Akhra , Dhaka"\n208,2023-10-30,10:30:00,52,126,109,26,859,"Shonir Akhra , Dhaka"\n209,2023-10-30,10:45:00,53,125,107,27,862,"Shonir Akhra , Dhaka"\n210,2023-10-
30,10:46:00,54,126,109,28,859,"Shonir Akhra , Dhaka"\n211,2023-10-30,10:47:00,56,124,111,29,861,"Shonir Akhra , Dhaka"\n212,2023-10-30,10:50:00,57,125,112,28,864,"Shonir Akhra , Dhaka"\n213,2023-10-
30,11:10:00,58,127,114,29,861,"Shonir Akhra , Dhaka"\n214,2023-10-31,12:46:00,57,125,115,30,862,"Shonir Akhra , Dhaka"\n215,2023-10-31,12:48:00,55,127,117,29,860,"Shonir Akhra , Dhaka"\n216,2023-10-
31,12:50:00,53,126,115,30,863,"Shonir Akhra , Dhaka"\n217,2023-10-31,12:51:00,54,128,114,31,862,"Shonir Akhra , Dhaka"\n218,2023-11-01,14:15:00,56,129,116,30,860,"Shonir Akhra , Dhaka"\n219,2023-11-
01,14:17:00,58,128,114,29,859,"Shonir Akhra , Dhaka"\n220,2023-11-01,14:20:00,60,126,116,28,860,"Shonir Akhra , Dhaka"\n221,2023-10-29,11:05:00,48,119,107,26,859,"Mohammadepur , Dhaka"\n222,2023-10-
29,12:45:00,49,120,109,27,860,"Mohammadepur , Dhaka"\n223,2023-10-30,10:20:00,51,121,111,26,858,"Mohammadepur , Dhaka"\n224,2023-10-30,10:25:00,52,122,112,27,859,"Mohammadepur , Dhaka"\n225,2023-10-
30,10:30:00,54,124,110,28,862,"Mohammadepur , Dhaka"\n226,2023-10-30,10:45:00,55,125,108,29,860,"Mohammadepur , Dhaka"\n227,2023-10-30,10:46:00,53,123,106,28,862,"Mohammadepur , Dhaka"\n228,2023-10-
30,10:47:00,54,124,107,27,863,"Mohammadepur , Dhaka"\n229,2023-10-30,10:50:00,52,125,109,28,861,"Mohammadepur , Dhaka"\n230,2023-10-30,11:10:00,53,124,110,27,858,"Mohammadepur , Dhaka"\n231,2023-10-
31,12:46:00,54,125,111,26,859,"Mohammadepur , Dhaka"\n232,2023-10-31,12:48:00,53,126,112,27,862,"Mohammadepur , Dhaka"\n233,2023-10-31,12:50:00,54,127,113,26,865,"Mohammadepur , Dhaka"\n234,2023-10-
31,12:51:00,55,128,115,27,863,"Mohammadepur , Dhaka"\n235,2023-11-01,14:15:00,53,127,113,26,862,"Mohammadepur , Dhaka"\n236,2023-11-01,14:17:00,54,125,115,27,859,"Mohammadepur , Dhaka"\n237,2023-11-
01,14:20:00,55,126,113,28,862,"Mohammadepur , Dhaka"\n238,2023-10-29,11:05:00,48,119,107,26,859,"Chow Rasta , Gazipur"\n239,2023-10-29,12:45:00,50,120,108,27,861,"Chow Rasta , Gazipur"\n240,2023-10-
30,10:20:00,52,122,106,26,859,"Chow Rasta , Gazipur"\n241,2023-10-30,10:25:00,50,120,108,27,861,"Chow Rasta , Gazipur"\n242,2023-10-30,10:30:00,48,121,107,28,858,"Chow Rasta , Gazipur"\n243,2023-10-
30,10:45:00,49,123,109,27,859,"Chow Rasta , Gazipur"\n244,2023-10-30,10:46:00,51,122,111,26,862,"Chow Rasta , Gazipur"\n245,2023-10-30,10:47:00,53,124,112,27,860,"Chow Rasta , Gazipur"\n246,2023-10-
30,10:50:00,52,126,111,28,858,"Chow Rasta , Gazipur"\n247,2023-10-30,11:10:00,53,127,112,29,861,"Chow Rasta , Gazipur"\n248,2023-10-31,12:46:00,54,129,114,30,864,"Chow Rasta , Gazipur"\n249,2023-10-
31,12:48:00,52,130,116,29,866,"Chow Rasta , Gazipur"\n250,2023-10-31,12:50:00,53,129,115,30,865,"Chow Rasta , Gazipur"\n251,2023-10-31,12:51:00,51,128,117,29,867,"Chow Rasta , Gazipur"\n252,2023-11-
01,14:15:00,49,129,116,28,870,"Chow Rasta , Gazipur"\n253,2023-11-01,14:17:00,47,131,115,27,867,"Chow Rasta , Gazipur"\n254,2023-11-01,14:20:00,48,133,114,26,865,"Chow Rasta , Gazipur"\n255,2023-10-
29,11:05:00,48,119,107,26,859,"Dhamrai , Dhaka"\n256,2023-10-29,12:45:00,50,121,109,27,862,"Dhamrai , Dhaka"\n257,2023-10-30,10:20:00,52,122,110,28,859,"Dhamrai , Dhaka"\n258,2023-10-30,10:25:00,53,124,108,29,862,"Dhamrai ,
Dhaka"\n259,2023-10-30,10:30:00,54,126,107,30,859,"Dhamrai , Dhaka"\n260,2023-10-30,10:45:00,52,128,109,31,862,"Dhamrai , Dhaka"\n261,2023-10-30,10:46:00,54,130,111,30,864,"Dhamrai , Dhaka"\n262,2023-10-
30,10:47:00,56,131,109,31,867,"Dhamrai , Dhaka"\n263,2023-10-30,10:50:00,55,133,111,30,866,"Dhamrai , Dhaka"\n264,2023-10-30,11:10:00,53,135,113,29,867,"Dhamrai , Dhaka"\n265,2023-10-31,12:46:00,55,136,115,30,865,"Dhamrai ,
Dhaka"\n266,2023-10-31,12:48:00,57,137,116,29,866,"Dhamrai , Dhaka"\n267,2023-10-31,12:50:00,59,138,115,30,865,"Dhamrai , Dhaka"\n268,2023-10-31,12:51:00,60,140,117,29,868,"Dhamrai , Dhaka"\n269,2023-11-
01,14:15:00,58,142,115,28,866,"Dhamrai , Dhaka"\n270,2023-11-01,14:17:00,60,143,117,29,867,"Dhamrai , Dhaka"\n271,2023-11-01,14:20:00,62,144,118,30,868,"Dhamrai , Dhaka"\n272,2023-10-29,11:05:00,48,119,107,26,859,"Gabtoli ,
```

```
Dhaka"\n273,2023-10-29,12:45:00,50,121,108,27,860,"Gabtoli , Dhaka"\n274,2023-10-30,10:20:00,49,119,109,28,861,"Gabtoli , Dhaka"\n275,2023-10-30,10:25:00,51,121,107,27,858,"Gabtoli , Dhaka"\n276,2023-10-
30,10:30:00,52,123,109,26,860,"Gabtoli , Dhaka"\n277,2023-10-30,10:45:00,53,124,108,27,858,"Gabtoli , Dhaka"\n278,2023-10-30,10:46:00,54,123,107,28,861,"Gabtoli , Dhaka"\n279,2023-10-30,10:47:00,55,124,108,27,864,"Gabtoli ,
Dhaka"\n280,2023-10-30,10:50:00,57,125,106,28,861,"Gabtoli , Dhaka"\n281,2023-10-30,11:10:00,59,126,108,27,862,"Gabtoli , Dhaka"\n282,2023-10-31,12:46:00,57,127,107,28,860,"Gabtoli , Dhaka"\n283,2023-10-
31,12:48:00,55,126,109,27,859,"Gabtoli , Dhaka"\n284,2023-10-31,12:50:00,53,124,107,28,861,"Gabtoli , Dhaka"\n285,2023-10-31,12:51:00,54,123,109,27,863,"Gabtoli , Dhaka"\n286,2023-11-01,14:15:00,55,124,111,26,861,"Gabtoli ,
Dhaka"\n287,2023-11-01,14:17:00,54,125,112,27,859,"Gabtoli , Dhaka"\n288,2023-11-01,14:20:00,55,126,110,26,862,"Gabtoli , Dhaka"\n289,2023-10-29,11:05:00,48,119,107,26,859,"Uttara Diyabari , Dhaka"\n290,2023-10-
29,12:45:00,49,120,108,27,860,"Uttara Diyabari , Dhaka"\n291,2023-10-30,10:20:00,50,119,107,28,857,"Uttara Diyabari , Dhaka"\n292,2023-10-30,10:25:00,48,120,108,29,859,"Uttara Diyabari , Dhaka"\n293,2023-10-
30,10:30:00,50,121,109,28,861,"Uttara Diyabari , Dhaka"\n294,2023-10-30,10:45:00,51,123,111,27,858,"Uttara Diyabari , Dhaka"\n295,2023-10-30,10:46:00,53,125,112,28,859,"Uttara Diyabari , Dhaka"\n296,2023-10-
30,10:47:00,52,127,113,27,862,"Uttara Diyabari , Dhaka"\n297,2023-10-30,10:50:00,54,125,112,28,864,"Uttara Diyabari , Dhaka"\n298,2023-10-30,11:10:00,55,127,113,29,865,"Uttara Diyabari , Dhaka"\n299,2023-10-
31,12:46:00,56,128,112,30,863,"Uttara Diyabari , Dhaka"\n300,2023-10-31,12:48:00,57,130,113,29,865,"Uttara Diyabari , Dhaka"\n301,2023-10-31,12:50:00,55,131,114,28,867,"Uttara Diyabari , Dhaka"\n302,2023-10-
31,12:51:00,56,133,116,27,868,"Uttara Diyabari , Dhaka"\n303,2023-11-01,14:15:00,58,135,117,26,870,"Uttara Diyabari , Dhaka"\n304,2023-11-01,14:17:00,57,137,119,27,869,"Uttara Diyabari , Dhaka"\n305,2023-11-
01,14:20:00,56,139,121,26,867,"Uttara Diyabari , Dhaka"\n306,2023-10-29,11:05:00,48,119,107,26,859,"Aftabnogor , Dhaka"\n307,2023-10-29,12:45:00,49,120,108,27,860,"Aftabnogor , Dhaka"\n308,2023-10-
30,10:20:00,51,121,110,26,857,"Aftabnogor , Dhaka"\n309,2023-10-30,10:25:00,52,123,108,27,860,"Aftabnogor , Dhaka"\n310,2023-10-30,10:30:00,50,125,110,28,857,"Aftabnogor , Dhaka"\n311,2023-10-
30,10:45:00,51,126,112,29,858,"Aftabnogor , Dhaka"\n312,2023-10-30,10:46:00,49,128,114,28,861,"Aftabnogor , Dhaka"\n313,2023-10-30,10:47:00,50,130,116,27,859,"Aftabnogor , Dhaka"\n314,2023-10-
30,10:50:00,52,132,117,26,862,"Aftabnogor , Dhaka"\n315,2023-10-30,11:10:00,50,133,116,27,859,"Aftabnogor , Dhaka"\n316,2023-10-31,12:46:00,48,135,114,26,860,"Aftabnogor , Dhaka"\n317,2023-10-
31,12:48:00,49,136,115,27,857,"Aftabnogor , Dhaka"\n318,2023-10-31,12:50:00,51,138,117,28,858,"Aftabnogor , Dhaka"\n319,2023-10-31,12:51:00,53,140,116,27,860,"Aftabnogor , Dhaka"\n320,2023-11-
01,14:15:00,54,142,115,28,857,"Aftabnogor , Dhaka"\n321,2023-11-01,14:17:00,53,140,117,29,860,"Aftabnogor , Dhaka"\n322,2023-11-01,14:20:00,55,142,119,28,858,"Aftabnogor , Dhaka"\n323,2023-10-29,11:05:00,48,119,107,26,859,"Bosila,
Dhaka"\n324,2023-10-29,12:45:00,50,121,108,27,862,"Bosila, Dhaka"\n325,2023-10-30,10:20:00,52,119,109,28,861,"Bosila, Dhaka"\n326,2023-10-30,10:25:00,53,121,111,29,858,"Bosila, Dhaka"\n327,2023-10-
30,10:30:00,54,122,109,30,860,"Bosila, Dhaka"\n328,2023-10-30,10:45:00,52,124,111,29,857,"Bosila, Dhaka"\n329,2023-10-30,10:46:00,50,125,109,28,858,"Bosila, Dhaka"\n330,2023-10-30,10:47:00,51,123,111,27,860,"Bosila,
Dhaka"\n331,2023-10-30,10:50:00,50,121,112,26,858,"Bosila, Dhaka"\n332,2023-10-30,11:10:00,49,122,113,27,859,"Bosila, Dhaka"\n333,2023-10-31,12:46:00,51,124,115,28,862,"Bosila, Dhaka"\n334,2023-10-
31,12:48:00,49,123,117,27,865,"Bosila, Dhaka"\n335,2023-10-31,12:50:00,51,124,119,26,867,"Bosila, Dhaka"\n336,2023-10-31,12:51:00,52,126,120,27,870,"Bosila, Dhaka"\n337,2023-11-01,14:15:00,53,125,118,26,873,"Bosila,
Dhaka"\n338,2023-11-01,14:17:00,55,126,119,27,875,"Bosila, Dhaka"\n339,2023-11-01,14:20:00,53,128,121,28,877,"Bosila, Dhaka"\n340,2023-10-29,11:05:00,48,119,107,26,859,"Mirpur-12 , Dhaka"\n341,2023-10-
29,12:45:00,50,120,108,27,861,"Mirpur-12 , Dhaka"\n342,2023-10-30,10:20:00,52,121,110,26,864,"Mirpur-12 , Dhaka"\n343,2023-10-30,10:25:00,54,120,112,27,862,"Mirpur-12 , Dhaka"\n344,2023-10-30,10:30:00,55,119,111,26,865,"Mirpur-12 ,
Dhaka"\n345,2023-10-30,10:45:00,57,121,113,27,866,"Mirpur-12 , Dhaka"\n346,2023-10-30,10:46:00,58,120,112,26,867,"Mirpur-12 , Dhaka"\n347,2023-10-30,10:47:00,56,121,114,27,865,"Mirpur-12 , Dhaka"\n348,2023-10-
30,10:50:00,57,123,112,26,864,"Mirpur-12 , Dhaka"\n349,2023-10-30,11:10:00,59,124,113,27,865,"Mirpur-12 , Dhaka"\n350,2023-10-31,12:46:00,60,125,111,26,863,"Mirpur-12 , Dhaka"\n351,2023-10-31,12:48:00,62,126,112,27,866,"Mirpur-12 ,
Dhaka"\n352,2023-10-31,12:50:00,64,124,113,28,869,"Mirpur-12 , Dhaka"\n353,2023-10-31,12:51:00,66,125,114,27,868,"Mirpur-12 , Dhaka"\n354,2023-11-01,14:15:00,68,126,115,28,870,"Mirpur-12 , Dhaka"\n355,2023-11-
01,14:17:00,70,127,116,29,871,"Mirpur-12 , Dhaka"\n356,2023-11-01,14:20:00,69,126,117,28,872,"Mirpur-12 , Dhaka"\n"}
```


```
[148] df = pd.read_csv('agriculture.csv')
```

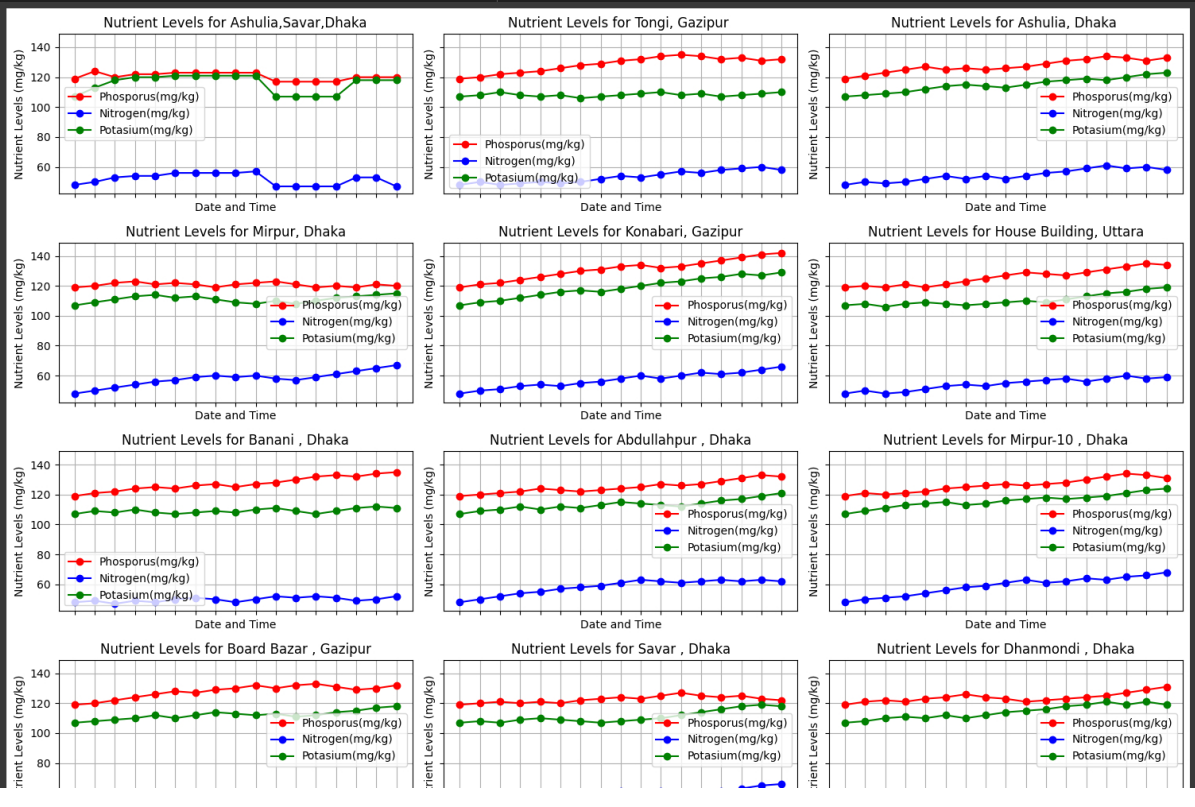
WE CREATED 357 ROWS (DUMMY DATA) USING THE GIVEN ONLY 17 DATA:

```
[149] print("Dataset Shape =", df.shape)
      print("\nDataset Overview with all possible values")

      print(df.columns)
```

```
Dataset Shape = (357, 9)

Dataset Overview with all possible values
Index(['Unnamed: 0', 'Date', 'Time', 'Nitrogen(mg/kg)', 'Phosporus(mg/kg)',
       'Potasium(mg/kg)', 'Temperature', 'Moisture', 'Location'],
      dtype='object')
```

VISUALIZATION: Location based plot that represents how soil reacts with providing 100gm of fertilizer in terms of soil nutrients

```
[150] locations = []
      locations = df['Location'].unique()

      num_subplots = len(locations)
      num_rows = (num_subplots - 1)// 3 + 1
      num_cols = min(3, num_subplots)
      fig, axs = plt.subplots(num_rows, num_cols, figsize=(15,20), sharex=True, sharey=True)
      axs = axs.flatten()


      for i, location in enumerate(locations):
          ax = axs[i]
          data = df[df['Location'] == location]
          ax.plot(data['Date']+ '' + data['Time'], data['Phosporus(mg/kg)'],label='Phosporus(mg/kg)', color='red', marker='o')
          ax.plot(data['Date']+ '' + data['Time'], data['Nitrogen(mg/kg)'],label='Nitrogen(mg/kg)', color='blue', marker='o')
          ax.plot(data['Date']+ '' + data['Time'], data['Potasium(mg/kg)'],label='Potasium(mg/kg)', color='green', marker='o')

          ax.set_title(f'Nutrient Levels for {location}')
          ax.set_ylabel('Nutrient Levels (mg/kg)')
          ax.legend()
          ax.grid(True)

          ax.set_xlabel('Date and Time')
          ax.tick_params(axis='x', rotation=90)

      plt.tight_layout()
      if num_subplots < num_rows*num_cols:
          for i in range(num_subplots, num_rows * num_cols):
              fig.delaxes(axs[i])
      plt.show()
```



FEATURE EXTRACTION: We calculate Nutrients level change compared to initial value(before fertilizer was added) for each observation

```
[151] df['Change in Nitrogen'] = df['Nitrogen(mg/kg)'] - df['Nitrogen(mg/kg)'].iloc[0]
      df['Change in Phosporus'] = df['Phosporus(mg/kg)'] - df['Phosporus(mg/kg)'].iloc[0]
      df['Change in Potasium'] = df['Potasium(mg/kg)'] - df['Potasium(mg/kg)'].iloc[0]

      print(df.columns)
```

```
Index(['Unnamed: 0', 'Date', 'Time', 'Nitrogen(mg/kg)', 'Phosporus(mg/kg)',
       'Potasium(mg/kg)', 'Temperature', 'Moisture', 'Location',
       'Change in Nitrogen', 'Change in Phosporus', 'Change in Potasium'],
      dtype='object')
```

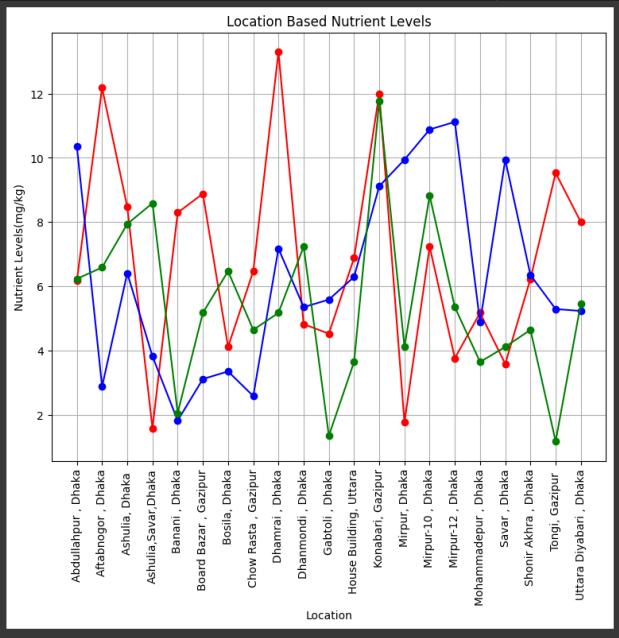
ANALYSIS: New dataframe that contains average of location based nutrients level wheather data, meaning for 20 location we get 20 observations or rows

```
[152] grouped = df.groupby('Location')
      means = grouped[['Change in Nitrogen','Change in Phosporus','Change in Potasium','Temperature','Moisture']].mean()

      means.reset_index(inplace=True)
      agg_df = means.copy()
      print(agg_df)
```

```
                    Location  Change in Nitrogen  Change in Phosporus  \
0       Abdullahpur , Dhaka           10.352941             6.176471
1        Aftabnogor , Dhaka            2.882353            12.176471
2           Ashulia, Dhaka            6.411765             8.470588
3       Ashulia,Savar,Dhaka            3.823529             1.588235
4            Banani , Dhaka            1.823529             8.294118
5     Board Bazar , Gazipur            3.117647             8.882353
6             Bosila, Dhaka            3.352941             4.117647
7      Chow Rasta , Gazipur            2.588235             6.470588
8           Dhamrai , Dhaka            7.176471            13.294118
9         Dhanmondi , Dhaka            5.352941             4.823529
10          Gabtoli , Dhaka            5.588235             4.529412
11  House Building, Uttara             6.294118             6.882353
12        Konabari, Gazipur            9.117647            12.000000
13            Mirpur, Dhaka            9.941176             1.764706
14        Mirpur-10 , Dhaka           10.882353             7.235294
15        Mirpur-12 , Dhaka           11.117647             3.764706
16     Mohammadepur , Dhaka            4.882353             5.176471
17            Savar , Dhaka            9.941176             3.588235
18     Shonir Akhra , Dhaka            6.352941             6.235294
19          Tongi, Gazipur             5.294118             9.529412
20  Uttara Diyabari , Dhaka            5.235294             8.000000

    Change in Potasium  Temperature    Moisture
0             6.235294    27.058824  860.823529
1             6.588235    27.294118  858.941176
2             7.941176    28.235294  860.705882
3             8.588235    27.941176  862.647059
4             2.058824    27.882353  861.294118
5             5.176471    27.529412  860.529412
6             6.470588    27.411765  863.588235
7             4.647059    27.647059  862.470588
8             5.176471    29.176471  864.235294
9             7.235294    26.588235  862.823529
10            1.352941    27.058824  860.529412
11            3.647059    27.647059  864.705882
12           11.764706    27.294118  866.705882
13            4.117647    28.352941  861.588235
14            8.823529    27.411765  862.529412
15            5.352941    26.941176  865.705882
16            3.647059    27.058824  860.823529
17            4.117647    27.882353  860.294118
18            4.647059    28.352941  860.588235
19            1.176471    30.352941  866.117647
20            5.470588    27.647059  863.117647
```

Visualization: Plot that represents influence or change of nutrient level after 4-6 days of observations for each location

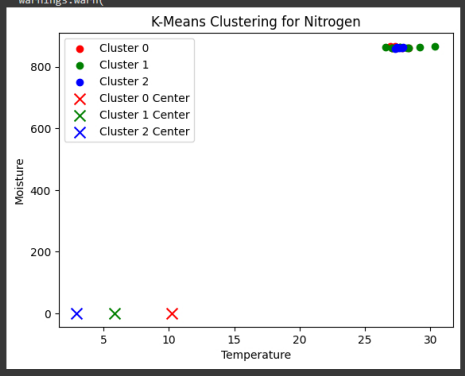
```
[153] plt.figure(figsize=(9,7))
      locations = agg_df['Location'].unique()
      x = range(len(locations))

      plt.plot (x, agg_df.groupby('Location')['Change in Phosporus'].mean(), label='Phosporus(mg/kg)',color='red',marker='o')
      plt.plot (x, agg_df.groupby('Location')['Change in Nitrogen'].mean(), label='Nitrogen(mg/kg)',color='blue',marker='o')
      plt.plot (x, agg_df.groupby('Location')['Change in Potasium'].mean(), label='Potassium(mg/kg)',color='green',marker='o')


      plt.title('Location Based Nutrient Levels')
      plt.xlabel('Location')
      plt.ylabel('Nutrient Levels(mg/kg)')
      plt.xticks(x, locations, rotation=90)
      plt.grid(True)
      plt.show()
```



CLUSTERING: K MEANS CLUSTERING FOR NITROGEN LEVEL

```
[154] X = agg_df['Change in Nitrogen'].values.reshape(-1,1)
      n_clusters = 3
      kmeans = KMeans(n_clusters = n_clusters, random_state=0).fit(X)
      cluster_labels = kmeans.labels_
      cluster_centers = kmeans.cluster_centers_

      agg_df['Nitrogen Cluster'] = cluster_labels
      cluster_means = agg_df.groupby('Nitrogen Cluster')['Change in Nitrogen'].mean()

      colors = ['r', 'g', 'b']
      for cluster in range(n_clusters):
          plt.scatter(agg_df[agg_df['Nitrogen Cluster'] == cluster]['Temperature'],
                      agg_df[agg_df['Nitrogen Cluster'] == cluster]['Moisture'],
                      label = f'Cluster {cluster}', c=colors[cluster])

      for cluster, center in enumerate(cluster_centers):
          plt.scatter(center, 0, label=f'Cluster {cluster} Center', marker='x', s=100, c=colors[cluster])

      plt.xlabel('Temperature')
      plt.ylabel('Moisture')
      plt.legend()
      plt.title('K-Means Clustering for Nitrogen')
      plt.show()

      print("\nCluster Means for Change in Nitrogen:")
      print(cluster_means)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

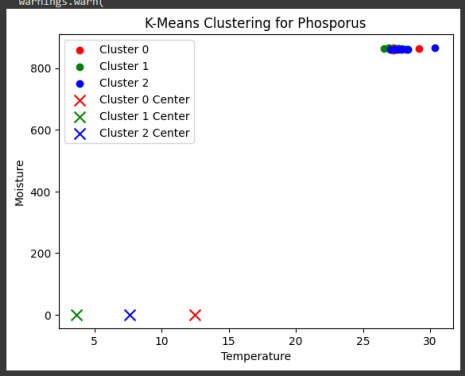



```
Cluster Means for Change in Nitrogen:
Nitrogen Cluster
0    10.225490
1     5.843137
2     2.931373
Name: Change in Nitrogen, dtype: float64
```

CLUSTERING: K MEANS CLUSTERING FOR PHOSPORUS LEVEL

```
[155] X = agg_df['Change in Phosporus'].values.reshape(-1,1)
      n_clusters = 3
      kmeans = KMeans(n_clusters = n_clusters, random_state=0).fit(X)
      cluster_labels = kmeans.labels_
      cluster_centers = kmeans.cluster_centers_

      agg_df['Phosporus Cluster'] = cluster_labels
      cluster_means = agg_df.groupby('Phosporus Cluster')['Change in Phosporus'].mean()

      colors = ['r', 'g', 'b']
      for cluster in range(n_clusters):
          plt.scatter(agg_df[agg_df['Phosporus Cluster'] == cluster]['Temperature'],
                      agg_df[agg_df['Phosporus Cluster'] == cluster]['Moisture'],
                      label = f'Cluster {cluster}', c=colors[cluster])

      for cluster, center in enumerate(cluster_centers):
          plt.scatter(center, 0, label=f'Cluster {cluster} Center', marker='x', s=100, c=colors[cluster])

      plt.xlabel('Temperature')
      plt.ylabel('Moisture')
      plt.legend()
      plt.title('K-Means Clustering for Phosporus')
      plt.show()

      print("\nCluster Means for Change in Phosporus:")
      print(cluster_means)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```



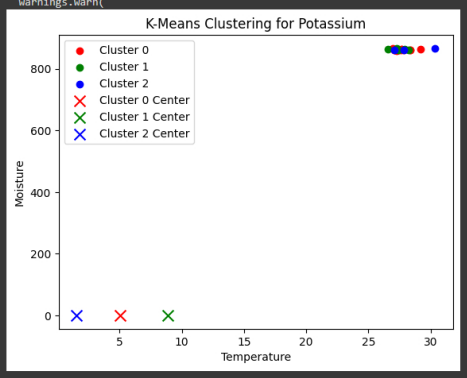

```
Cluster Means for Change in Phosporus:
Phosporus Cluster
0    12.490196
1     3.669118
2     7.617647
Name: Change in Phosporus, dtype: float64
```

CLUSTERING: K MEANS CLUSTERING FOR POTASSIUM

```
[156] X = agg_df['Change in Potasium'].values.reshape(-1,1)
```

```
n_clusters = 3
kmeans = KMeans(n_clusters = n_clusters, random_state=0).fit(X)
cluster_labels = kmeans.labels_
cluster_centers = kmeans.cluster_centers_

agg_df['Potasium Cluster'] = cluster_labels
cluster_means = agg_df.groupby('Potasium Cluster')['Change in Potasium'].mean()

colors = ['r', 'g', 'b']
for cluster in range(n_clusters):
    plt.scatter(agg_df[agg_df['Potasium Cluster'] == cluster]['Temperature'],
                agg_df[agg_df['Potasium Cluster'] == cluster]['Moisture'],
                label = f'Cluster {cluster}', c=colors[cluster])

for cluster, center in enumerate(cluster_centers):
    plt.scatter(center, 0, label=f'Cluster {cluster} Center', marker='x', s=100, c=colors[cluster])

plt.xlabel('Temperature')
plt.ylabel('Moisture')
plt.legend()
plt.title('K-Means Clustering for Potassium')
plt.show()

print("\nCluster Means for Change in Potassium:")
print(cluster_means)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

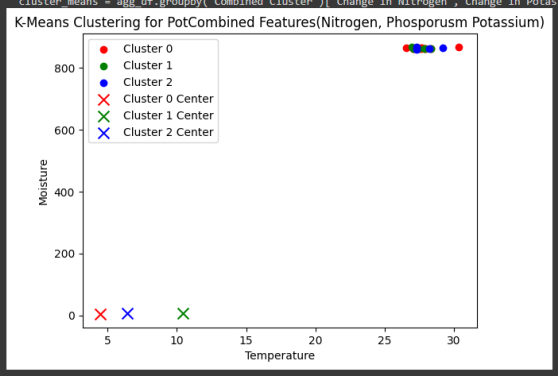



```
Cluster Means for Change in Potassium:
Potasium Cluster
0    5.022624
1    8.870588
2    1.529412
Name: Change in Potasium, dtype: float64
```

CLUSTERING K_MENS CLUSTERING FOR NITROGEN, PHOSPORUS AND POTASSIUM LEVEL COMBINED

```
[157] X = agg_df[['Change in Nitrogen','Change in Potasium','Change in Phosporus']]
n_clusters = 3

kmeans = KMeans(n_clusters = n_clusters, random_state=0).fit(X)

cluster_labels = kmeans.labels_
cluster_centers = kmeans.cluster_centers_

agg_df['Combined Cluster'] = cluster_labels
cluster_means = agg_df.groupby('Combined Cluster')['Change in Nitrogen','Change in Potasium','Change in Phosporus'].mean()

colors = ['r', 'g', 'b']
for cluster in range(n_clusters):
    plt.scatter(agg_df[agg_df['Combined Cluster'] == cluster]['Temperature'],
                agg_df[agg_df['Combined Cluster'] == cluster]['Moisture'],
                label = f'Cluster {cluster}', c=colors[cluster])

for cluster, center in enumerate(cluster_centers):
    plt.scatter(center[0],center[1], label=f'Cluster {cluster} Center', marker='x', s=100, c=colors[cluster])

plt.xlabel('Temperature')
plt.ylabel('Moisture')
plt.legend()
plt.title('K-Means Clustering for PotCombined Features(Nitrogen, Phosporusm Potassium)')
plt.show()

print("\nCluster Means for Combined Features:")
print(cluster_means)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
<ipython-input-157-ef1b1573fec1>:10: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
  cluster_means = agg_df.groupby('Combined Cluster')['Change in Nitrogen','Change in Potasium','Change in Phosporus'].mean()
```



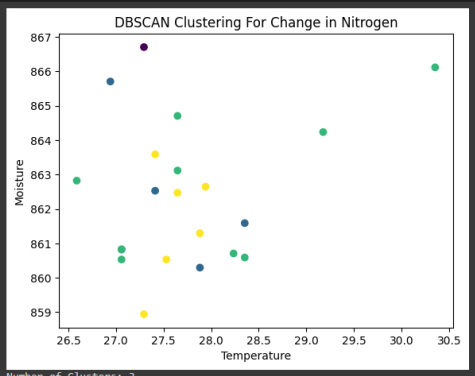

```
Cluster Means for Combined Features:
                  Change in Nitrogen  Change in Potasium  Change in Phosporus
Combined Cluster
0                           4.475490            4.509804             6.210784
1                          10.447059            5.729412             4.505882
2                           6.397059            7.867647            11.485294
```

Download the clustered dataset

```
[158] agg_df.to_csv('agricultureof.csv', index=False)
files.download('agricultureof.csv')
```

CLUSTERING: DBSCAN CLUSTERING FOR NITROGEN LEVEL

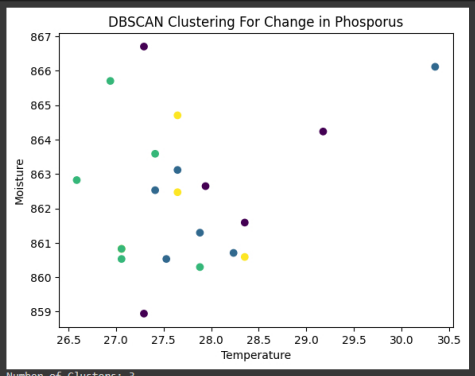
```
[159] X= agg_df[['Change in Nitrogen']].values
      scaler = StandardScaler()
      X_scaled = scaler.fit_transform(X)

      dbscan = DBSCAN(eps=0.3, min_samples=5).fit(X_scaled)

      agg_df['Nitrogen DBSCAN Cluster']= dbscan.labels_

      plt.scatter(agg_df['Temperature'], agg_df['Moisture'], c=agg_df['Nitrogen DBSCAN Cluster'])
      plt.xlabel('Temperature')
      plt.ylabel('Moisture')
      plt.title('DBSCAN Clustering For Change in Nitrogen')
      plt.show()

      n_clusters = len(set(dbscan.labels_)) - (1 if -1 in dbscan.labels_ else 0)
      n_noise = list(dbscan.labels_).count(-1)
      print(f'Number of Clusters: {n_clusters}')
      print(f'Number of Noise Points: {n_noise}')
```




```
Number of Clusters: 3
Number of Noise Points: 1
```

CLUSTERING: DBSCAN CLUSTERING FOR PHOSPORUS LEVEL

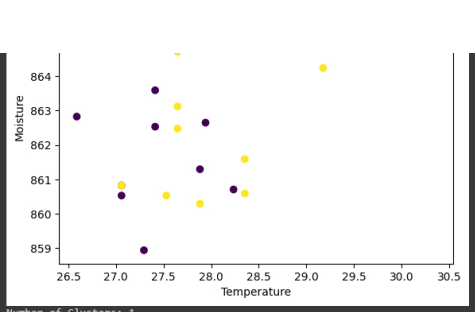
```
[160] X= agg_df[['Change in Phosporus']].values
      scaler = StandardScaler()
      X_scaled = scaler.fit_transform(X)

      dbscan = DBSCAN(eps=0.3, min_samples=5).fit(X_scaled)

      agg_df['Phosporus DBSCAN Cluster']= dbscan.labels_

      plt.scatter(agg_df['Temperature'], agg_df['Moisture'], c=agg_df['Phosporus DBSCAN Cluster'])
      plt.xlabel('Temperature')
      plt.ylabel('Moisture')
      plt.title('DBSCAN Clustering For Change in Phosporus')
      plt.show()

      n_clusters = len(set(dbscan.labels_)) - (1 if -1 in dbscan.labels_ else 0)
      n_noise = list(dbscan.labels_).count(-1)
      print(f'Number of Clusters: {n_clusters}')
      print(f'Number of Noise Points: {n_noise}')
```




```
Number of Clusters: 3
Number of Noise Points: 5
```

CLUSTERING: DBSCAN CLUSTERING FOR POTASSIUM LEVEL

```
X= agg_df[['Change in Potasium']].values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

dbscan = DBSCAN(eps=0.3, min_samples=5).fit(X_scaled)

agg_df['Potasium DBSCAN Cluster']= dbscan.labels_

plt.scatter(agg_df['Temperature'], agg_df['Moisture'], c=agg_df['Potasium DBSCAN Cluster'])
plt.xlabel('Temperature')
plt.ylabel('Moisture')
plt.title('DBSCAN Clustering For Change in Potassium')
plt.show()

n_clusters = len(set(dbscan.labels_)) - (1 if -1 in dbscan.labels_ else 0)
n_noise = list(dbscan.labels_).count(-1)
print(f'Number of Clusters: {n_clusters}')
print(f'Number of Noise Points: {n_noise}')
```




```
Number of Clusters: 1
Number of Noise Points: 11
```