Problem Statement

In the experiment, we took PES (commercially available medium) in which we have replaced nitrate and phosphate.

Now for Nitrate source it was replaced with Urea and for phosphate it was replaced with DAP, also we have check the combined

Effect of Urea and DAP as a source of nitrate and phosphate respectively.

Initial weight of the Ulva in triplicate given below and increased weight on 21st day (also we have given for 7th and 14th day increased weight in mg). this biomass was grown in 250ml of medium

Now we have to do prediction of If we could scale up for 20000 litre tank what will be production output per day per cycle (21 days) per year

When we altered medium

Condition 1. What happens when if we only replace with Urea in the medium? (asper the data provided)

Condition 2. What happens when if we only replace with DAP in the medium? (asper the data provided)

Condition 3. What happens when if we replace combining Urea and DAP? (asper the data provided)

Ulva Prediction

```
# import necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import csv
from sklearn.linear_model import LinearRegression
import warnings
warnings.filterwarnings("ignore")

# Required Functions

def extrapolate(x, y, day):
    y = y.astype(np.float)
    fit = np.polyfit(x, np.log(y), 1)
    temp = fit[1] + (fit[0] * day)
    return math.exp(temp)

def forecast( x, y, i):
    return  [extrapolate(x, y, j) for j in range(365)]
# Write forecast values into a file

def writeForecastIntoFile( filename, forecast):
    with open(filename, 'w', encoding='UTF8', newline='') as f:
        df = pd.read_csv("properties_file_days.csv")
        header = df.columns.tolist()
        writer = csv.writer(f)
        writer.writerow(header)
    with open(filename, 'a', encoding='UTF8', newline='') as f:
        writer = csv.writer(f)
        writer.writerows(forecast)

# Final Forecast in figures

def displayForecast(filename):
    # Take average and plot again
    average = []
    days = []

    average_cycle = []
    days_cycle = []

    df1 = pd.read_csv(filename)
    names = pd.read_csv("properties_file_days.csv")

    for i in range(len(names.columns)):
        val = df1[names.columns[i]].mean()
        # Scale it to 20000 Litre Tank
        val *= 80000
        average.append(val)
        days.append(i)

    for i in range(len(names.columns)):
        if (i%21)==0:
            val = df1[names.columns[i]].mean()
            # Scale it to 20000 Litre Tank
            val *= 80000
            average_cycle.append(val)
            days_cycle.append(i)


    print("Production at the end of the year: {:.2e} mg".format(average[364]))

    plt.scatter(days, average)
    plt.title("Ulva Production per day per Year")
    plt.xlabel("Days(0-365)")
    plt.ylabel("Production in (mg)")
    plt.show()

    plt.scatter(days_cycle, average_cycle)
    plt.title("Ulva Production per cycle per Year")
    plt.xlabel("Cycle( every 21 days)")
    plt.ylabel("Production in (mg)")
    plt.show()
```

# 1) Urea under consideration

```
urea = pd.read_csv('effect_of_urea.csv')
print(urea)

X = urea[["day_0", "day_7", "day_14"]]
y = urea["day_21"]
```

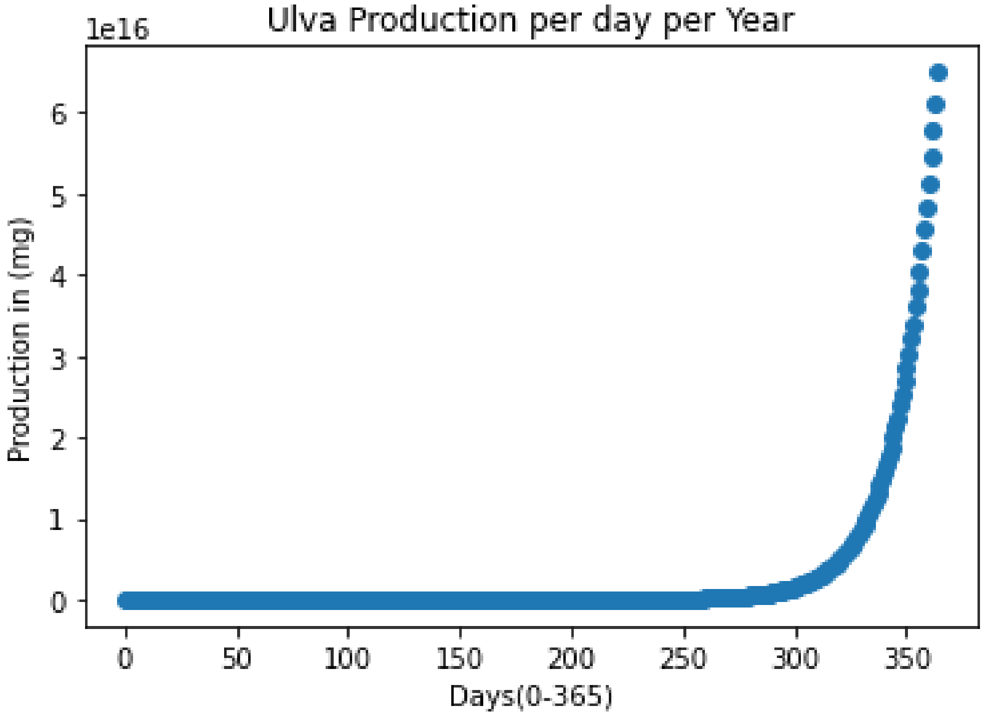
```
     medium    sample  day_0  day_7  day_14  day_21
```

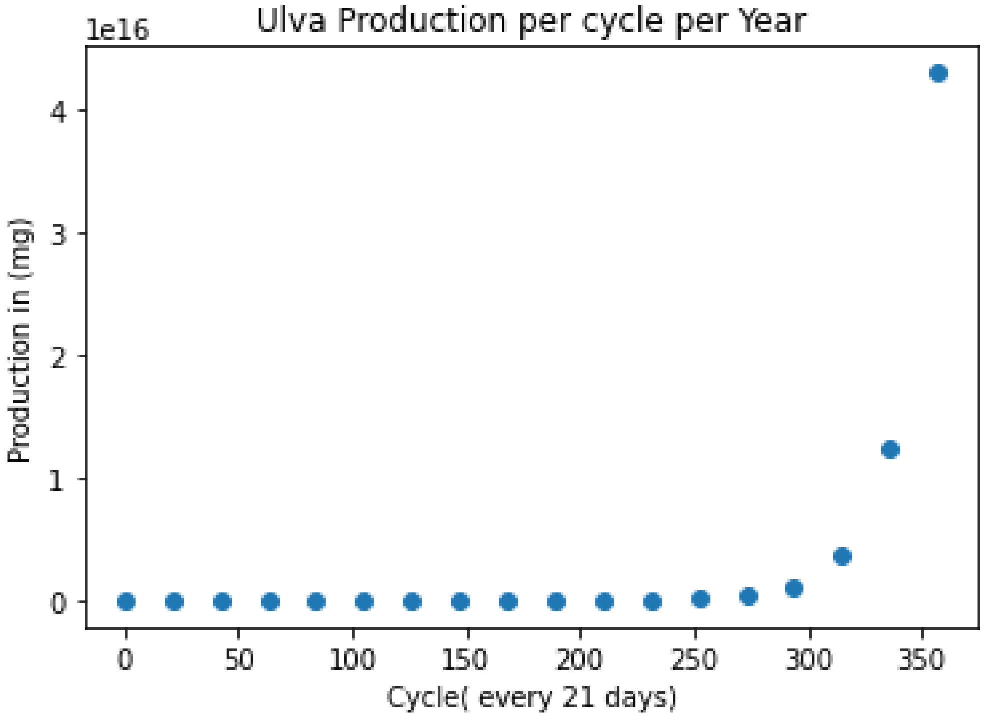
```
0   Control  growth_1  559.7  825.2  1367.3  1694.8
1   Control  growth_2  547.8  905.9  1493.1  1888.8
2   Control  growth_3  559.8  868.6  1454.0  1702.6
3     Lower  growth_1  460.3  709.5   994.5  1416.3
4     Lower  growth_2  496.5  700.5   976.5  1318.0
5     Lower  growth_3  475.1  974.5  1376.1  1830.3
6    Actual  growth_1  593.6  915.8  1518.3  2303.3
7    Actual  growth_2  483.0  910.5  1644.2  2222.0
8    Actual  growth_3  544.3  926.4  1632.4  2413.3
9    Higher  growth_1  596.5  794.4  1231.1  1728.0
10   Higher  growth_2  576.6  729.9  1236.5  1679.6
11   Higher  growth_3  525.6  928.2  1499.8  1986.7
```

```
# 1.1 Urea with Controlled concentration

header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [["Control","growth_1",559.7,825.2,1367.3,1694.8],
        ["Control","growth_2",547.8,905.9,1493.1,1888.8],
        ["Control","growth_3",559.8,868.6,1454,1702.6]]

# Write the model estimated output values to a csv file ('urea_control.csv')
with open('urea_control.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

urea_control = pd.read_csv('urea_control.csv')
print("urea_control: \n", urea_control)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(urea_control)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(urea_control.iloc[i, 2:6])

    a = forecast(x1, y1, i)
    forecast_list.append(a)

writeForecastIntoFile('urea_control_forecasted.csv', forecast_list )
displayForecast('urea_control_forecasted.csv')
```

```
urea_control:
     medium    sample  day_0  day_7  day_14  day_21
0  Control  growth_1  559.7  825.2  1367.3  1694.8
```

```
1  Control  growth_2  547.8  905.9  1493.1  1888.8
2  Control  growth_3  559.8  868.6  1454.0  1702.6
Production at the end of the year: 6.50e+16 mg
```



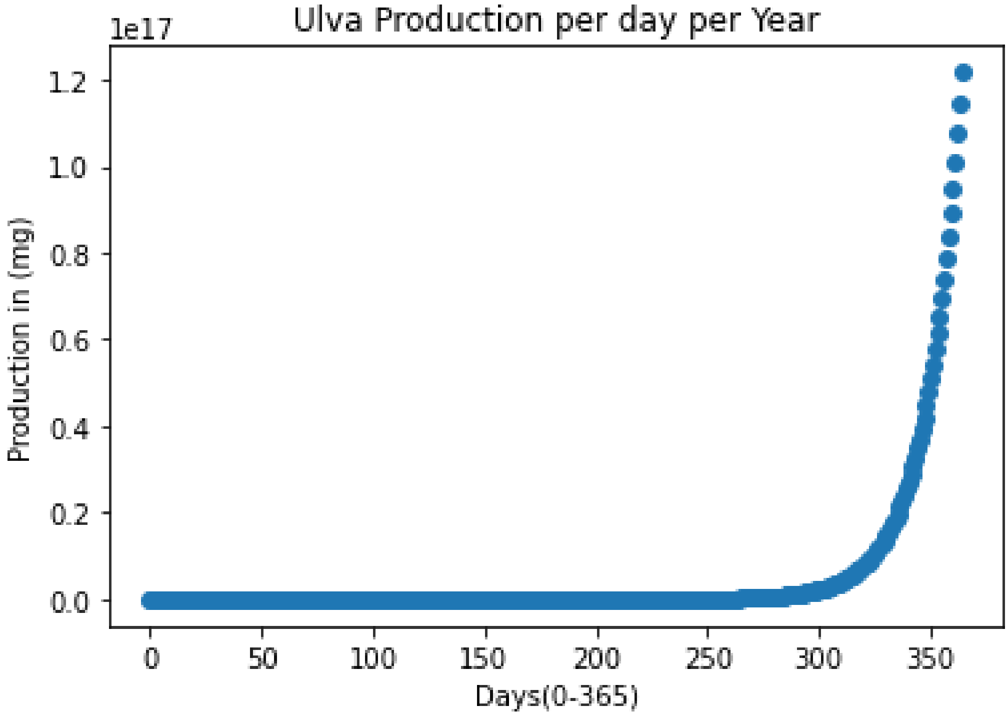


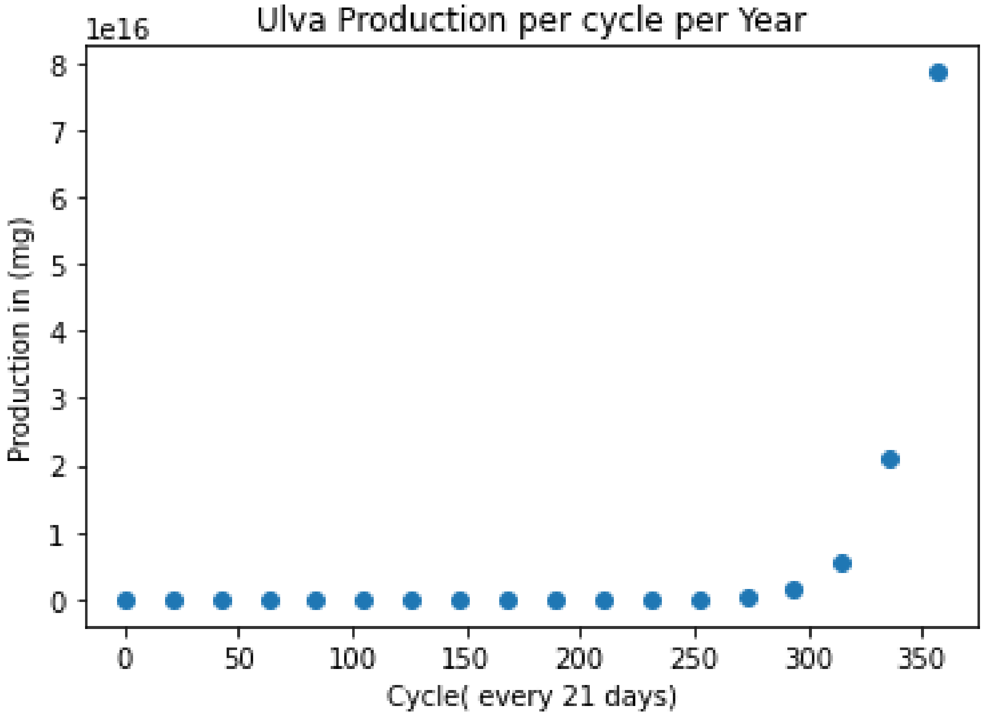
# 1.2 Urea with Lower concentration

header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Lower','growth_1',460.3,709.5,994.5,1416.3],

```
    ['Lower','growth_2',496.5,700.5,976.5,1318],
    ['Lower','growth_3',475.1,974.5,1376.1,1830.3]]

# Write the model estimated output values to a csv file ('urea_lower.csv')
with open('urea_lower.csv', 'w', encoding='UTF8', newline='') as f:
  writer = csv.writer(f)
  writer.writerow(header)
  writer.writerows(data)

# Confirming the updated data sample

urea_lower = pd.read_csv('urea_lower.csv')
print("urea_lower: \n", urea_lower)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(urea_lower)):
  x1 = np.arange(0, 22, 7)
  y1 = np.array(urea_lower.iloc[i, 2:6])

  a = forecast(x1, y1, i)
  forecast_list.append(a)

writeForecastIntoFile('urea_lower_forecasted.csv',forecast_list )
displayForecast('urea_lower_forecasted.csv')
```

```
urea_lower:
   medium    sample  day_0  day_7  day_14  day_21
0  Lower  growth_1  460.3  709.5   994.5  1416.3
1  Lower  growth_2  496.5  700.5   976.5  1318.0
2  Lower  growth_3  475.1  974.5  1376.1  1830.3
Production at the end of the year: 1.22e+17 mg
```

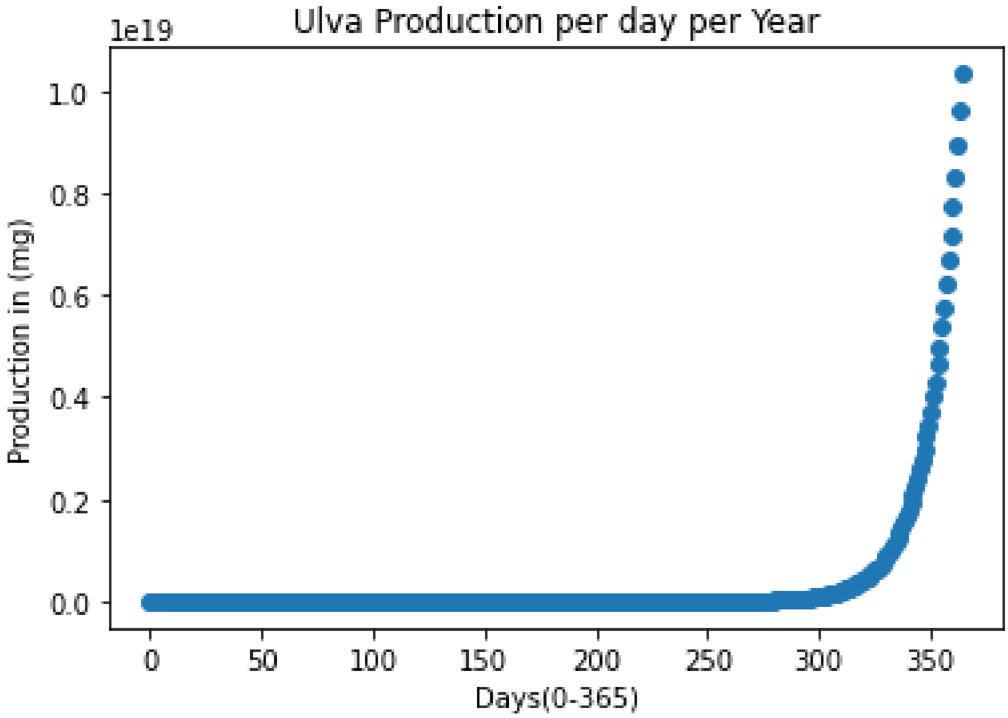


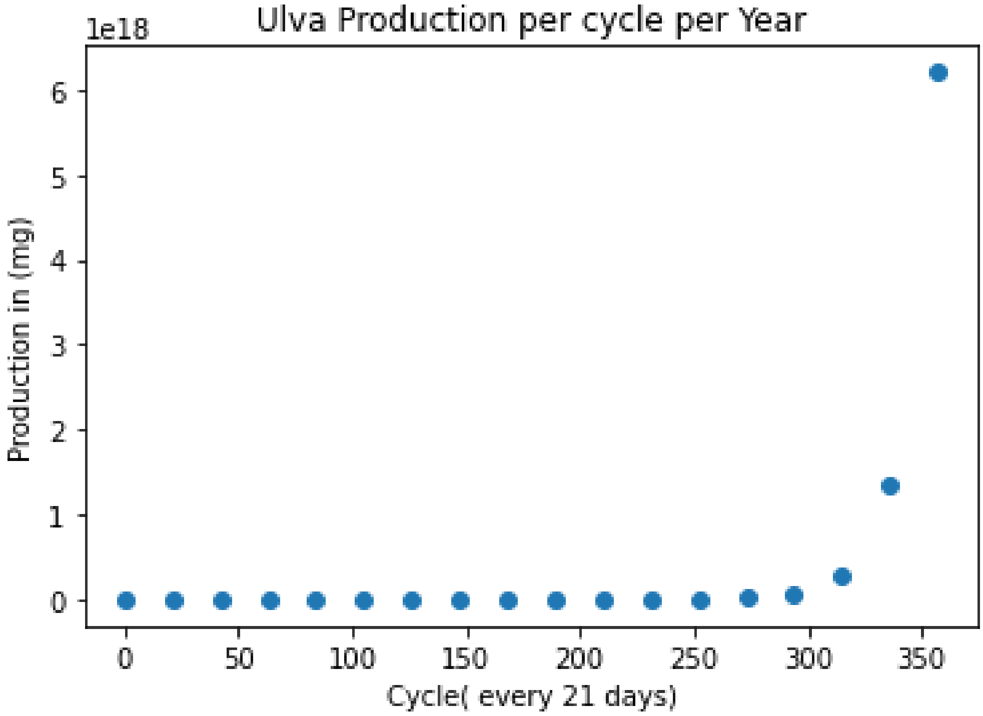


```
# 1.3 Urea with Actual concentration
header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Actual','growth_1',593.6,915.8,1518.3,2303.3],
      ['Actual','growth_2',483,910.5,1644.2,2222],
      ['Actual','growth_3',544.3,926.4,1632.4,2413.3]]

# Write the model estimated output values to a csv file ('urea_actual.csv')
with open('urea_actual.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

urea_actual = pd.read_csv('urea_actual.csv')
print("urea_actual: \n", urea_actual)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(urea_actual)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(urea_actual.iloc[i, 2:6])

    a = forecast(x1, y1, i)
    forecast_list.append(a)

writeForecastIntoFile('urea_actual_forecasted.csv',forecast_list )
displayForecast('urea_actual_forecasted.csv')
```

```
urea_actual:
     medium    sample  day_0  day_7  day_14  day_21
0  Actual  growth_1  593.6  915.8  1518.3  2303.3
1  Actual  growth_2  483.0  910.5  1644.2  2222.0
2  Actual  growth_3  544.3  926.4  1632.4  2413.3
```

```
Production at the end of the year: 1.03e+19 mg
```



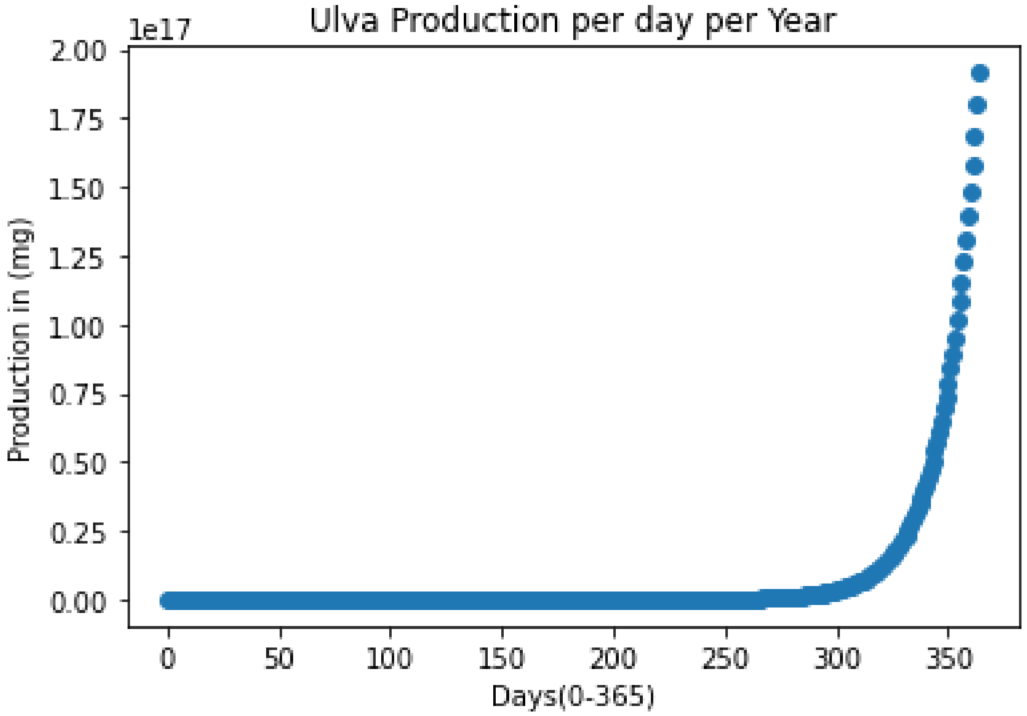


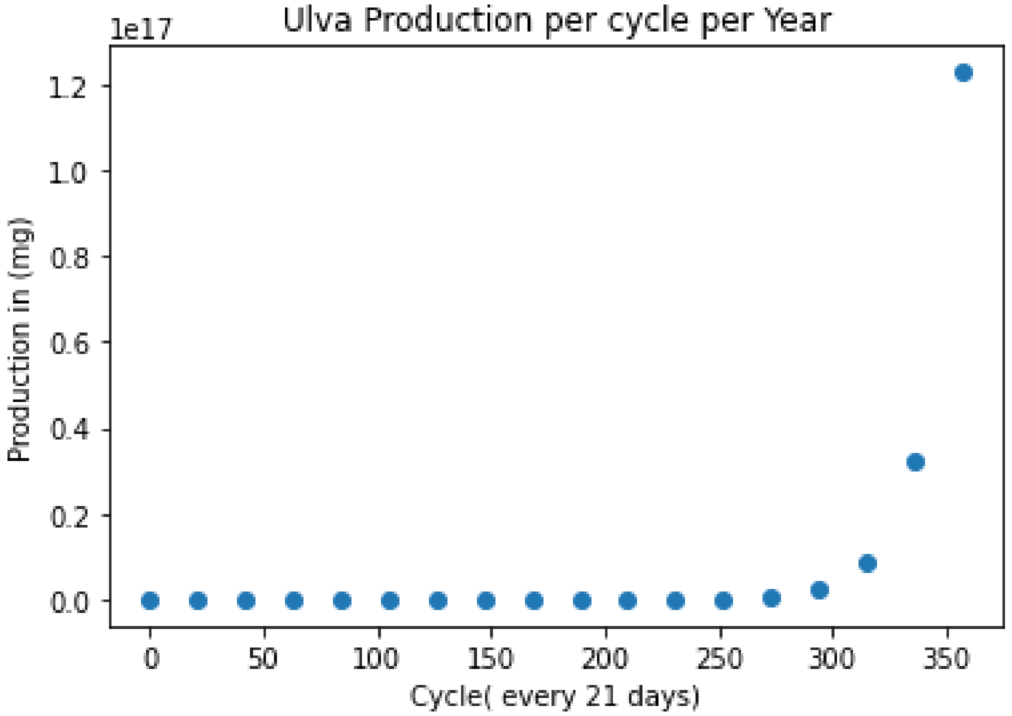
# 1.4 Urea with Higher concentration

```
header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Higher','growth_1',596.5,794.4,1231.1,1728],
        ['Higher','growth_2',576.6,729.9,1236.5,1679.6],
        ['Higher','growth_3',525.6,928.2,1499.8,1986.7]]
```

```
# Write the model estimated output values to a csv file ('urea_higher.csv')
with open('urea_higher.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

urea_higher = pd.read_csv('urea_higher.csv')
print("urea_higher: \n",urea_higher)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(urea_higher)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(urea_higher.iloc[i, 2:6])

    a = forecast(x1, y1, i)
    forecast_list.append(a)

writeForecastIntoFile('urea_higher_forecasted.csv',forecast_list )
displayForecast('urea_higher_forecasted.csv')
```

```
urea_higher:
    medium    sample  day_0  day_7  day_14  day_21
0  Higher  growth_1  596.5  794.4  1231.1  1728.0
1  Higher  growth_2  576.6  729.9  1236.5  1679.6
2  Higher  growth_3  525.6  928.2  1499.8  1986.7
Production at the end of the year: 1.92e+17 mg
```





# 2) DAP under consideration

```
DAP = pd.read_csv('effect_of_DAP.csv')
print(DAP)

X = DAP[["day_0", "day_7", "day_14"]]
y = DAP["day_21"]
```

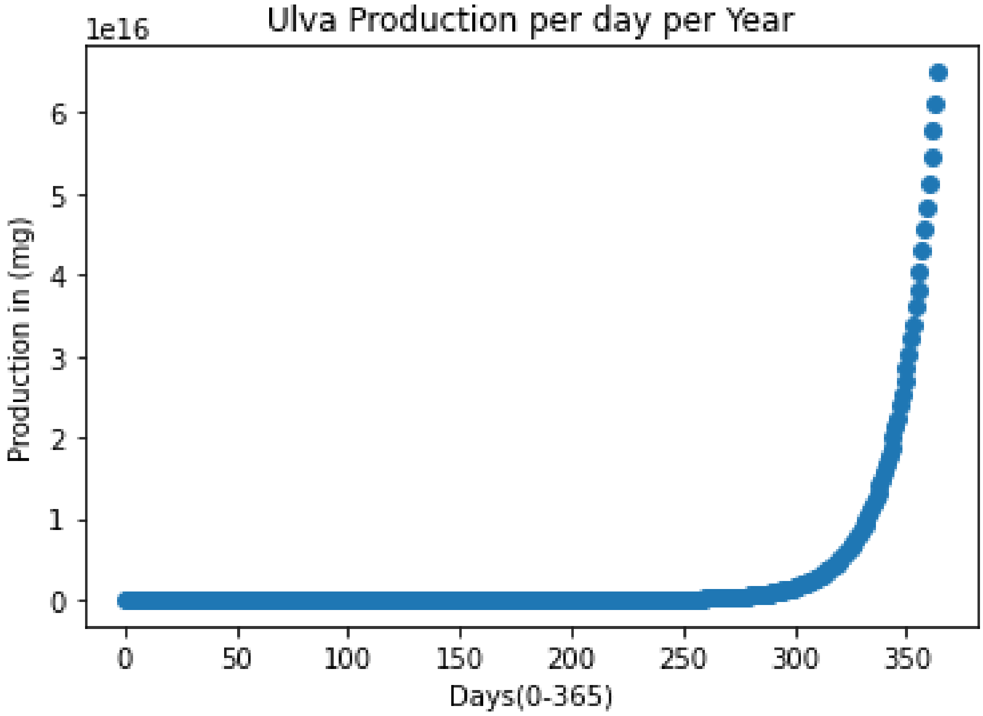
```
     medium    sample  day_0  day_7  day_14  day_21
```

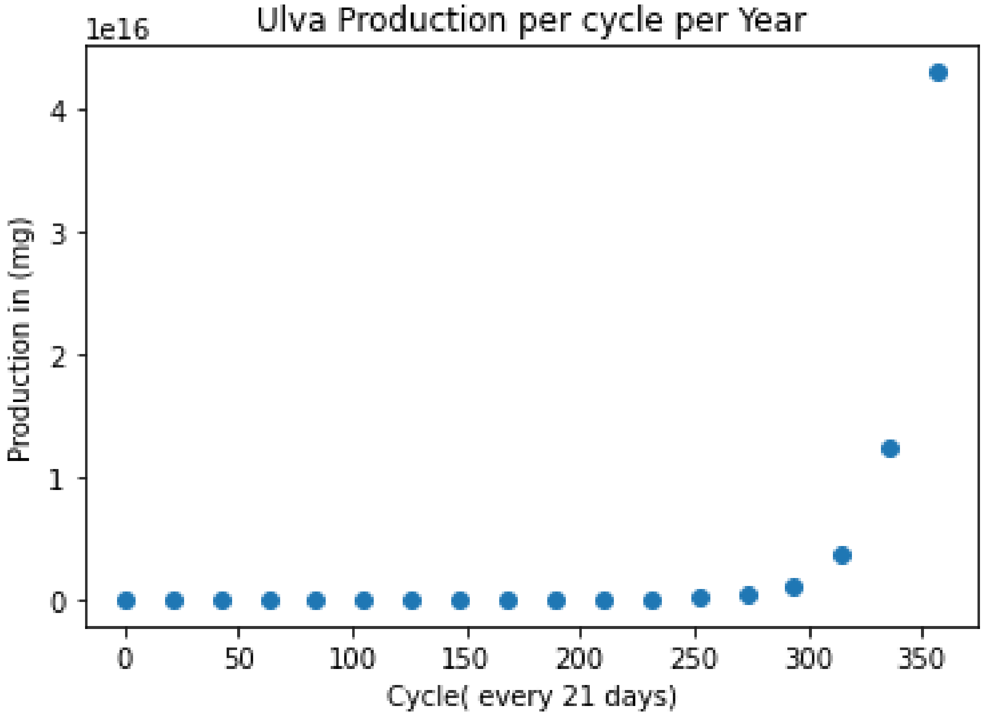
```
0   Control  growth_1  559.7  825.2  1367.3  1694.8
1   Control  growth_2  547.8  905.9  1493.1  1888.8
2   Control  growth_3  559.8  868.6  1454.0  1702.6
3     Lower  growth_1  529.9  760.3  1350.9  1558.2
4     Lower  growth_2  523.0  685.0  1133.4  1554.4
5     Lower  growth_3  383.3  480.8   660.9   903.8
6    Actual  growth_1  525.7  628.8  1010.2  1302.0
7    Actual  growth_2  488.2  653.8  1017.1  1370.4
8    Actual  growth_3  540.2  777.2  1226.2  1408.0
9    Higher  growth_1  435.9  468.4   807.0  1246.5
10   Higher  growth_2  387.8  446.2   753.4  1013.7
11   Higher  growth_3  438.2  463.6   815.2  1079.6
```

# 2.1 DAP with Controlled concentration

```
header = [“medium”,”sample”,”day_0”,”day_7”,”day_14”,”day_21”]
data = [[‘Control’,’growth_1’,559.7,825.2,1367.3,1694.8],
       [‘Control’,’growth_2’,547.8,905.9,1493.1,1888.8],
       [‘Control’,’growth_3’,559.8,868.6,1454,1702.6]]

# Write the model estimated output values to a csv file (‘DAP_control.csv’)
with open(‘DAP_control.csv’, ‘w’, encoding=’UTF8’, newline=’’) as f:
   writer = csv.writer(f)
   writer.writerow(header)
   writer.writerows(data)

# Confirming the updated data sample

DAP_control = pd.read_csv(‘DAP_control.csv’)
print(“DAP_control: \n”,DAP_control)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for I in range(len(DAP_control)):
   x1 = np.arange(0, 22, 7)
   y1 = np.array(DAP_control.iloc[I, 2:6])

   a = forecast(x1, y1, i)
   forecast_list.append(a)

writeForecastIntoFile(‘DAP_control_forecasted.csv’,forecast_list )
displayForecast(‘DAP_control_forecasted.csv’)
```

```
DAP_control:
     medium    sample  day_0  day_7  day_14  day_21
   2  Control  growth_1  559.7  825.2  1367.3  1694.8
   2  Control  growth_2  547.8  905.9  1493.1  1888.8
2  Control  growth_3  559.8  868.6  1454.0  1702.6
Production at the end of the year: 6.50e+16 mg
```

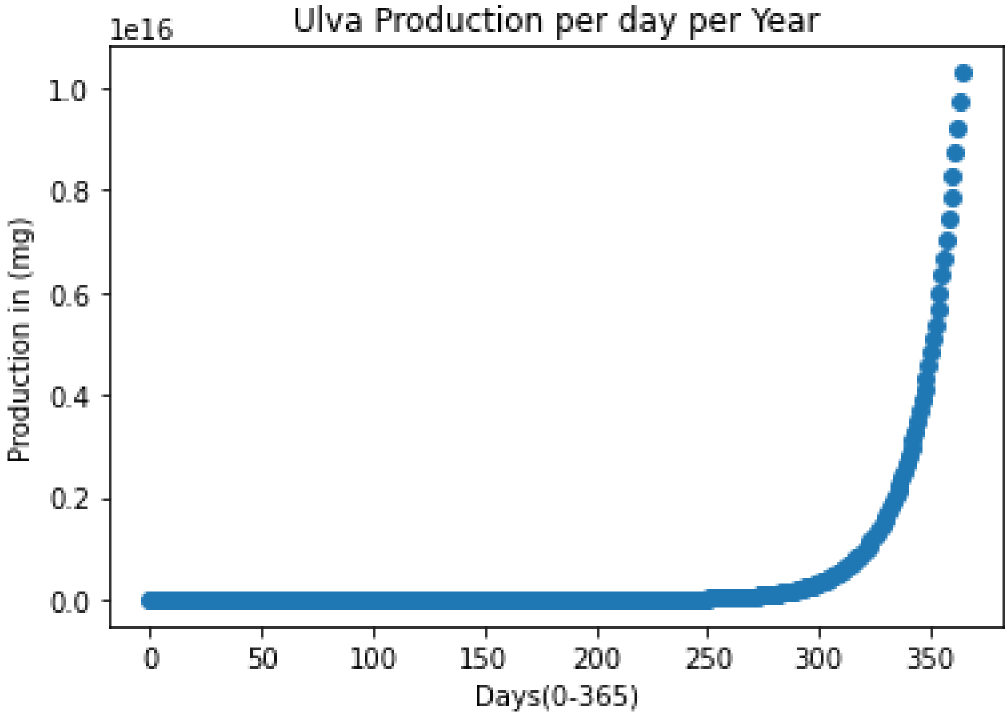


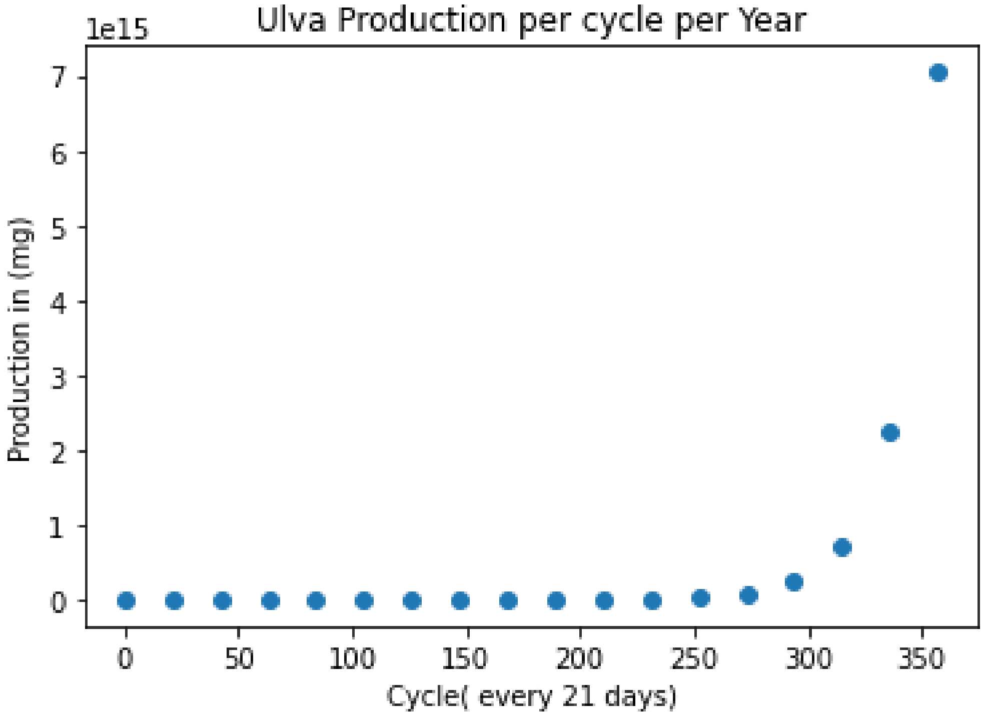


# 2.2 DAP with Lower concentration

```
header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Lower','growth_1',529.9,760.3,1350.9,1558.2],
        ['Lower','growth_2',523,685,1133.4,1554.4],
        ['Lower','growth_3',383.3,480.8,660.9,903.8]]

# Write the model estimated output values to a csv file ('DAP_lower.csv')
with open('DAP_lower.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

DAP_lower = pd.read_csv('DAP_lower.csv')
print("DAP_lower: \n",DAP_lower)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(DAP_lower)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(DAP_lower.iloc[i, 2:6])

    a = forecast(x1, y1, i)
    forecast_list.append(a)

writeForecastIntoFile('DAP_lower_forecasted.csv',forecast_list )
displayForecast('DAP_lower_forecasted.csv')
```

```
DAP_lower:
   medium    sample  day_0  day_7  day_14  day_21
0  Lower  growth_1  529.9  760.3  1350.9  1558.2
1  Lower  growth_2  523.0  685.0  1133.4  1554.4
2  Lower  growth_3  383.3  480.8   660.9   903.8
Production at the end of the year: 1.03e+16 mg
```

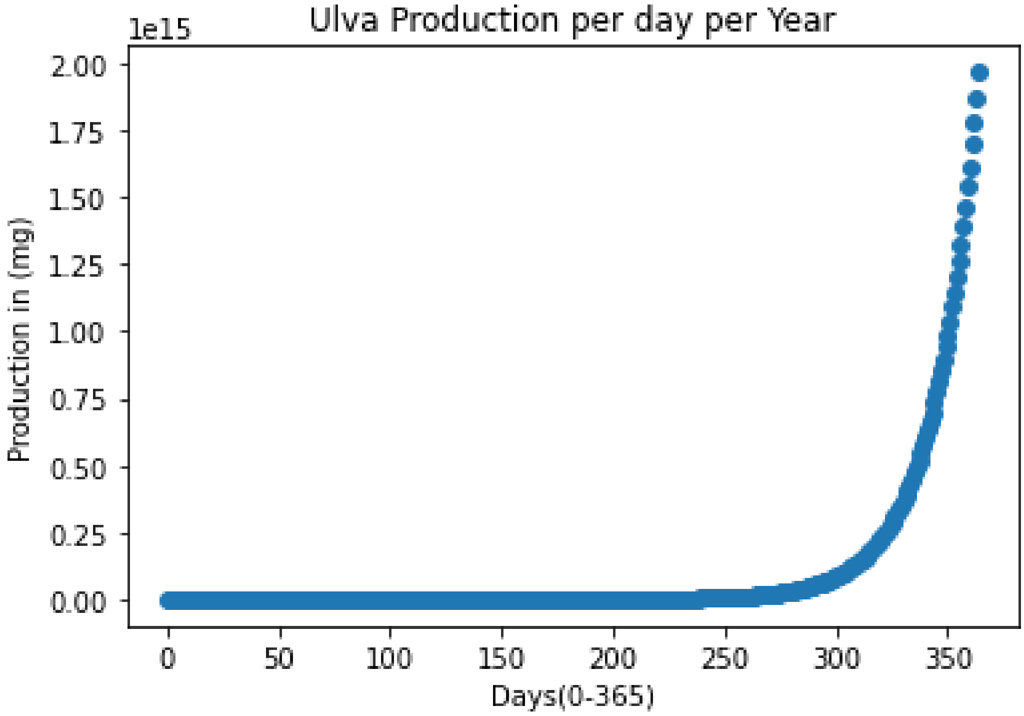


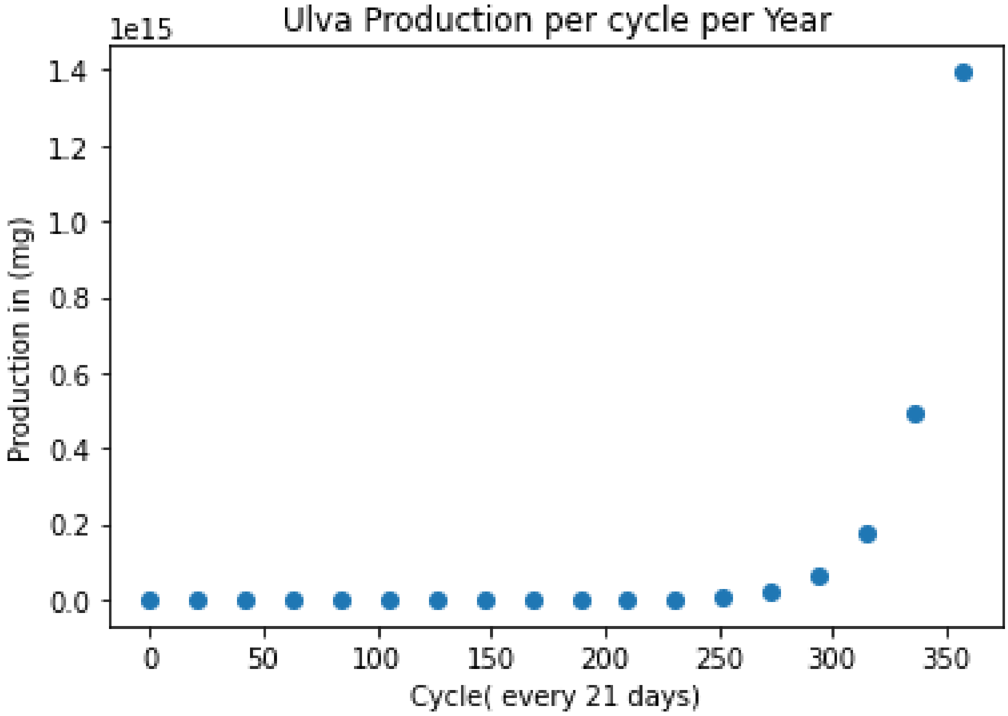


```
# 2.3 DAP with Actual concentration

header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Actual','growth_1',525.7,628.8,1010.2,1302],
        ['Actual','growth_2',488.2,653.8,1017.1,1370.4],
        ['Actual','growth_3',540.2,777.2,1226.2,1408]]

# Write the model estimated output values to a csv file ('DAP_actual.csv')
```

```
with open('DAP_actual.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

DAP_actual = pd.read_csv('DAP_actual.csv')
print("DAP_actual: \n", DAP_actual)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(DAP_actual)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(DAP_actual.iloc[i, 2:6])

    a = forecast(x1, y1, i)
    forecast_list.append(a)

writeForecastIntoFile('DAP_actual_forecasted.csv',forecast_list )
displayForecast('DAP_actual_forecasted.csv')
```

```
DAP_actual:
     medium    sample  day_0  day_7  day_14  day_21
0  Actual  growth_1  525.7  628.8  1010.2  1302.0
1  Actual  growth_2  488.2  653.8  1017.1  1370.4
2  Actual  growth_3  540.2  777.2  1226.2  1408.0
Production at the end of the year: 1.97e+15 mg
```



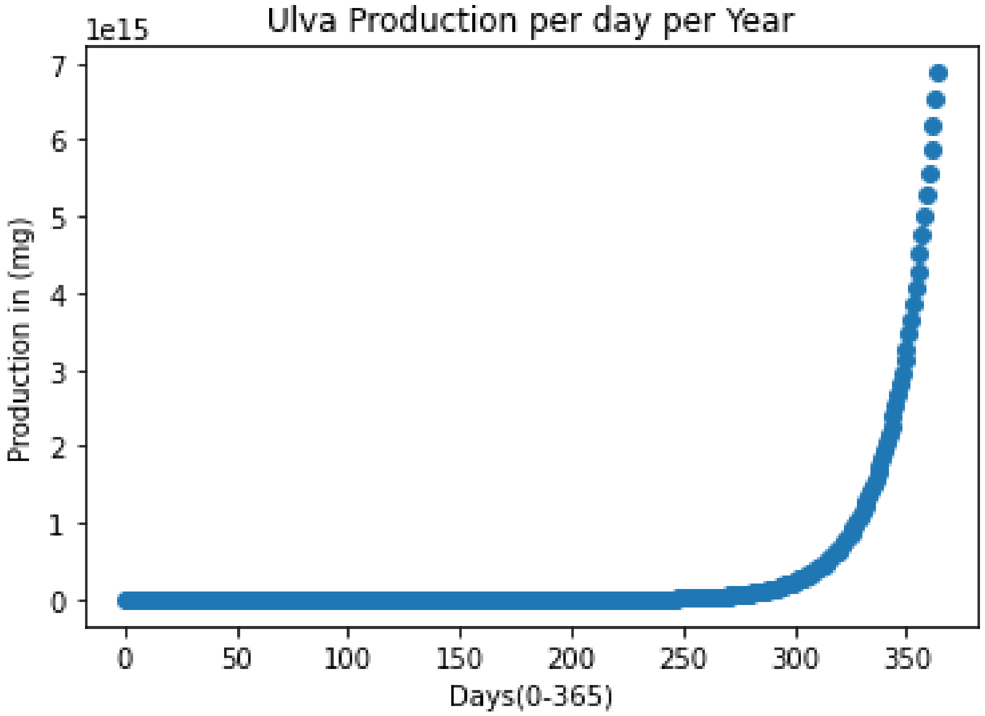


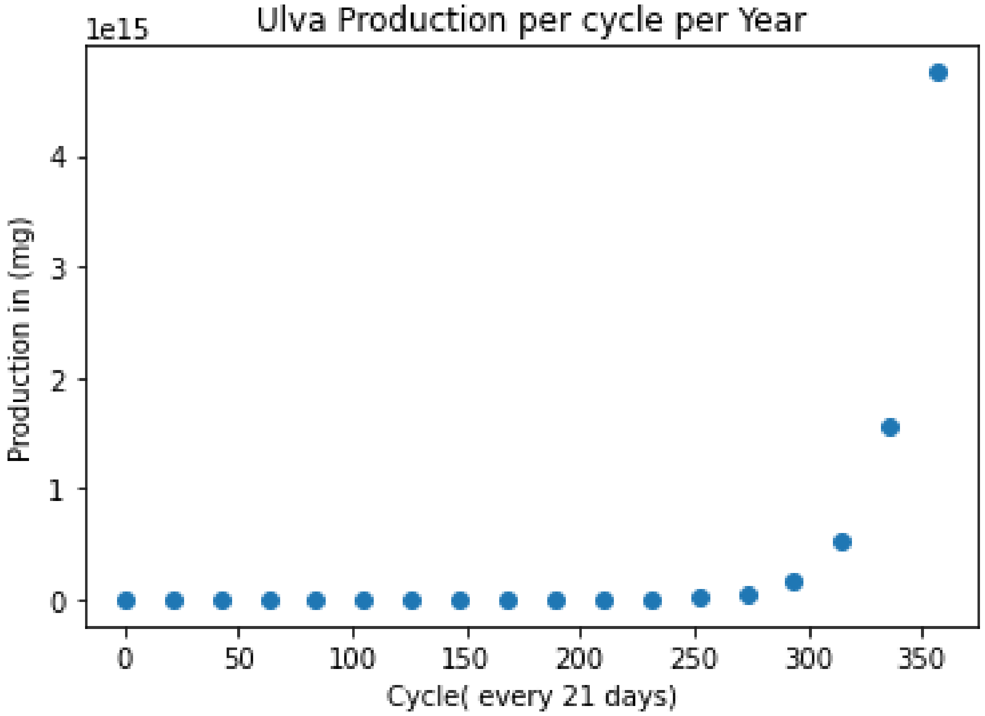
# 2.4 DAP with Higher concentration

```
header = [“medium”,”sample”,”day_0”,”day_7”,”day_14”,”day_21”]
data = [[‘Higher’,’growth_1’,435.9,468.4,807,1246.5],
       [‘Higher’,’growth_2’,387.8,446.2,753.4,1013.7],
       [‘Higher’,’growth_3’,438.2,463.6,815.2,1079.6]]

# Write the model estimated output values to a csv file (‘DAP_higher.csv’)
```

```
with open(‘DAP_higher.csv’, ‘w’, encoding=’UTF8’, newline=’’) as f:
   writer = csv.writer(f)
   writer.writerow(header)
   writer.writerows(data)

# Confirming the updated data sample

DAP_higher = pd.read_csv(‘DAP_higher.csv’)
print(“DAP_higher: \n”, DAP_higher)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for I in range(103):
   x1 = np.arange(0, 22, 7)
   y1 = np.array(DAP_higher.iloc[I, 2:6])

   try:
      a = forecast(x1, y1, i)
      forecast_list.append(a)
      writeForecastIntoFile(‘DAP_higher_forecasted.csv’,forecast_list )

      displayForecast(‘DAP_higher_forecasted.csv’)

   finally:
      print(“Cannot forecast due to negative extrapolation. Poor performance”)
      break
```

```
DAP_higher:
    medium    sample  day_0  day_7  day_14  day_21
0  Higher  growth_1  435.9  468.4   807.0  1246.5
1  Higher  growth_2  387.8  446.2   753.4  1013.7
2  Higher  growth_3  438.2  463.6   815.2  1079.6
Production at the end of the year: 6.88e+15 mg
```





Cannot forecast due to negative extrapolation. Poor performance

# 3) Both Urea and DAP under consideration

```
both = pd.read_csv('effect_of_both.csv')
print(both)

X = both[["day_0", "day_7", "day_14"]]
```

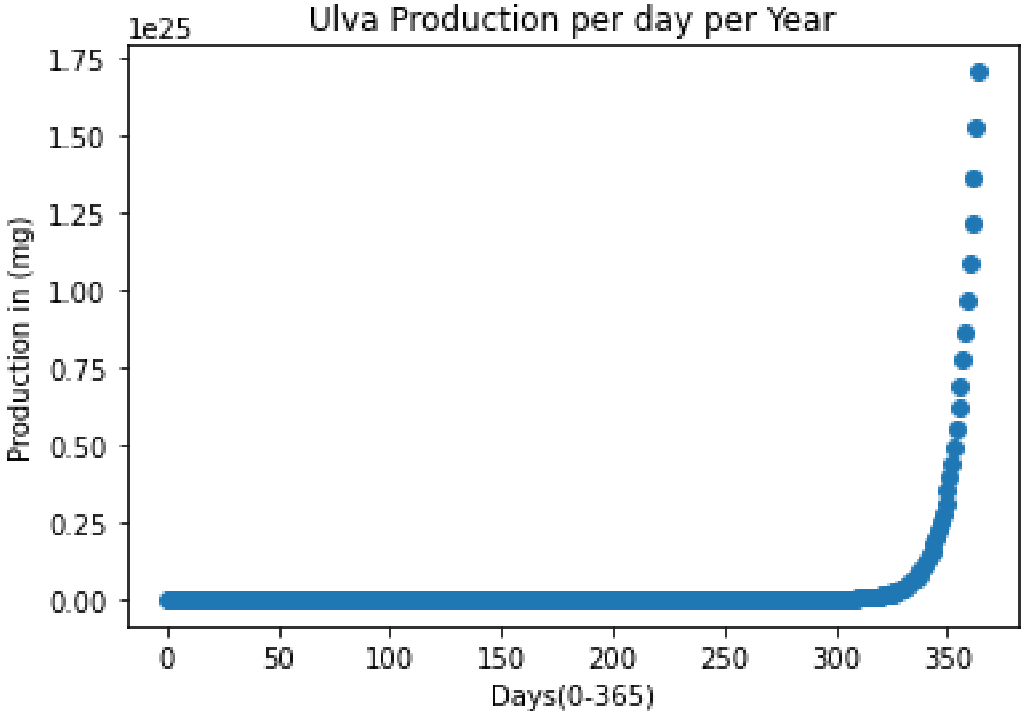
```
y = both["day_21"]
```

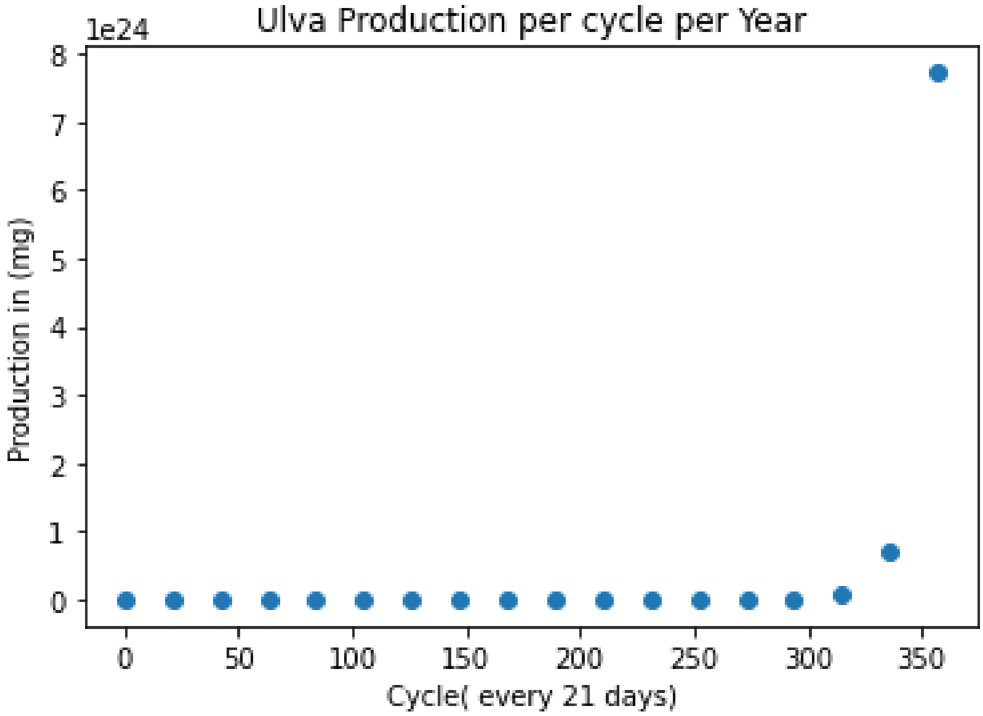
```
      medium    sample  day_0  day_7  day_14  day_21
0    Control  growth_1  349.2  528.7  1080.0  3849.0
1    Control  growth_2  382.6  747.2  1334.1  4167.0
2    Control  growth_3  349.1  754.9  1260.5  2090.5
3      Lower  growth_1  353.9  499.8   743.7  1055.1
4      Lower  growth_2  326.7  498.5   938.3  1353.6
5      Lower  growth_3  381.0  501.6   884.0  1467.5
6     Actual  growth_1  359.2  968.0  1368.9  2029.0
7     Actual  growth_2  439.6  811.6  1217.5  2121.0
8     Actual  growth_3  286.8  547.5   796.6  1181.9
9     Higher  growth_1  314.7  412.7   664.9   950.4
10    Higher  growth_2  241.4  322.1   685.2   801.5
11    Higher  growth_3  257.5  278.0   610.3   712.7
```

```
# 3.1 both with Controlled concentration
header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Control','growth_1',349.2,528.7,1080,3849],
        ['Control','growth_2',382.6,747.2,1334.1,4167],
        ['Control','growth_3',349.1,754.9,1260.5,2090.5]]

# Write the model estimated output values to a csv file ('both_control.csv')
with open('both_control.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

both_control = pd.read_csv('both_control.csv')
print("both_control: \n", both_control)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(both_control)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(both_control.iloc[i, 2:6])

    try:
        a = forecast(x1, y1, i)
        forecast_list.append(a)
        writeForecastIntoFile('both_control_forecasted.csv',forecast_list )

        displayForecast('both_control_forecasted.csv')
```

```
finally:
    print("Cannot forecast due to negative extrapolation. Poor performance")
    break
```

```
both_control:
     medium    sample  day_0  day_7  day_14  day_21
0  Control  growth_1  349.2  528.7  1080.0  3849.0
1  Control  growth_2  382.6  747.2  1334.1  4167.0
2  Control  growth_3  349.1  754.9  1260.5  2090.5
Production at the end of the year: 1.70e+25 mg
```

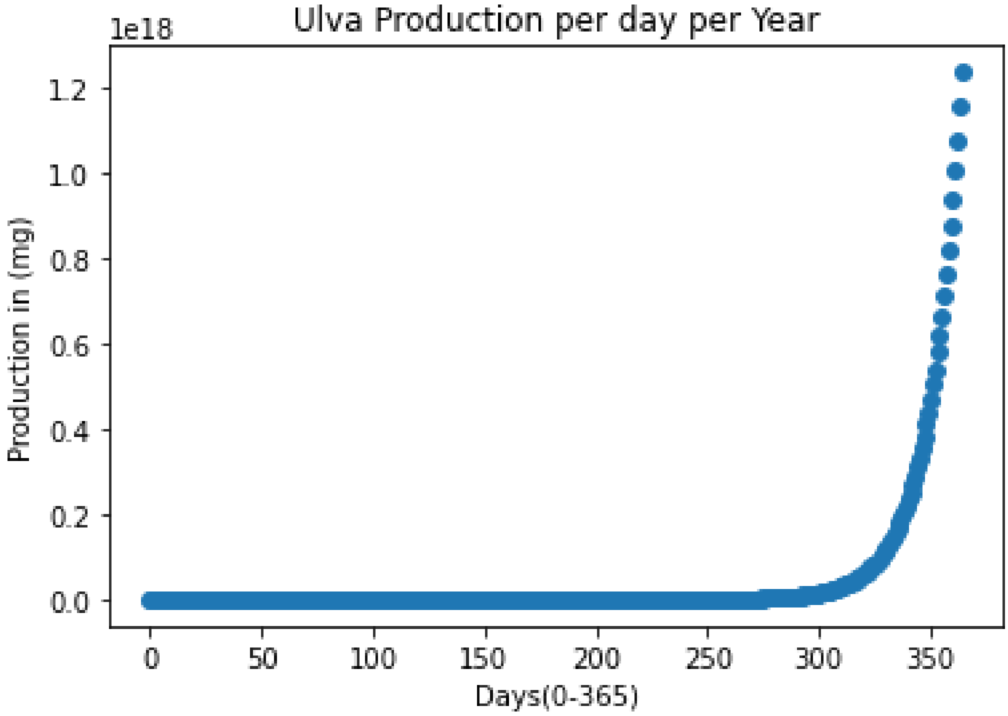


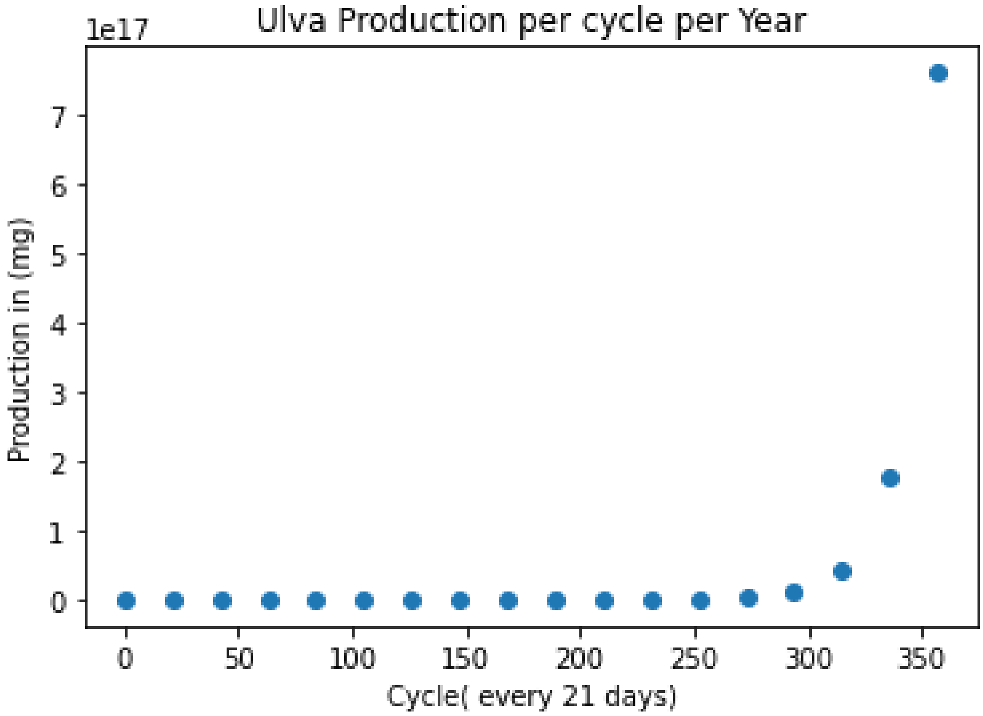


```
Cannot forecast due to negative extrapolation. Poor performance
```

```
# 3.2 both with Lower concentration

header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Lower','growth_1',353.9,499.8,743.7,1055.1],
       ['Lower','growth_2',326.7,498.5,938.3,1353.6],
       ['Lower','growth_3',381,501.6,884,1467.5]]

# Write the model estimated output values to a csv file ('both_lower.csv')
with open('both_lower.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

both_lower = pd.read_csv('both_lower.csv')
print("both_lower: \n", both_lower)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(both_lower)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(both_lower.iloc[i, 2:6])

    a = forecast(x1, y1, i)
    forecast_list.append(a)

writeForecastIntoFile('both_lower_forecasted.csv',forecast_list )
displayForecast('both_lower_forecasted.csv')
```

```
both_lower:
   medium    sample  day_0  day_7  day_14  day_21
0  Lower  growth_1  353.9  499.8   743.7  1055.1
1  Lower  growth_2  326.7  498.5   938.3  1353.6
2  Lower  growth_3  381.0  501.6   884.0  1467.5
Production at the end of the year: 1.24e+18 mg
```

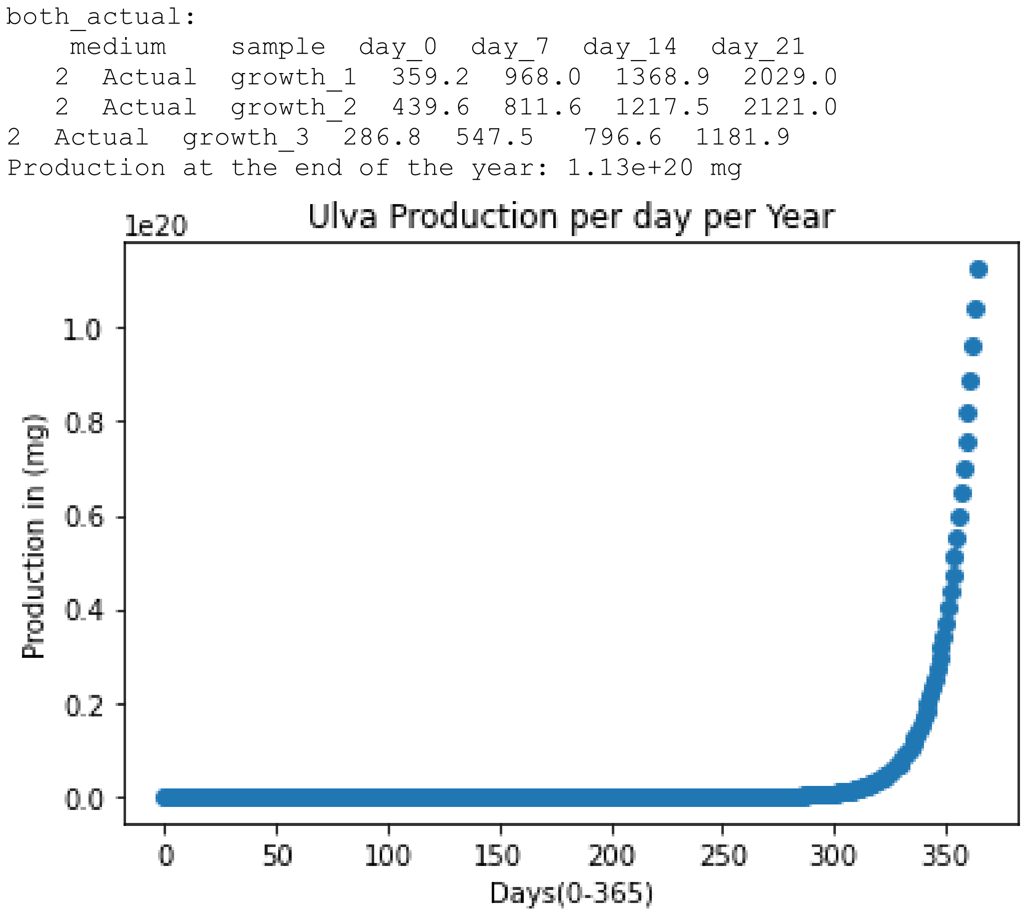




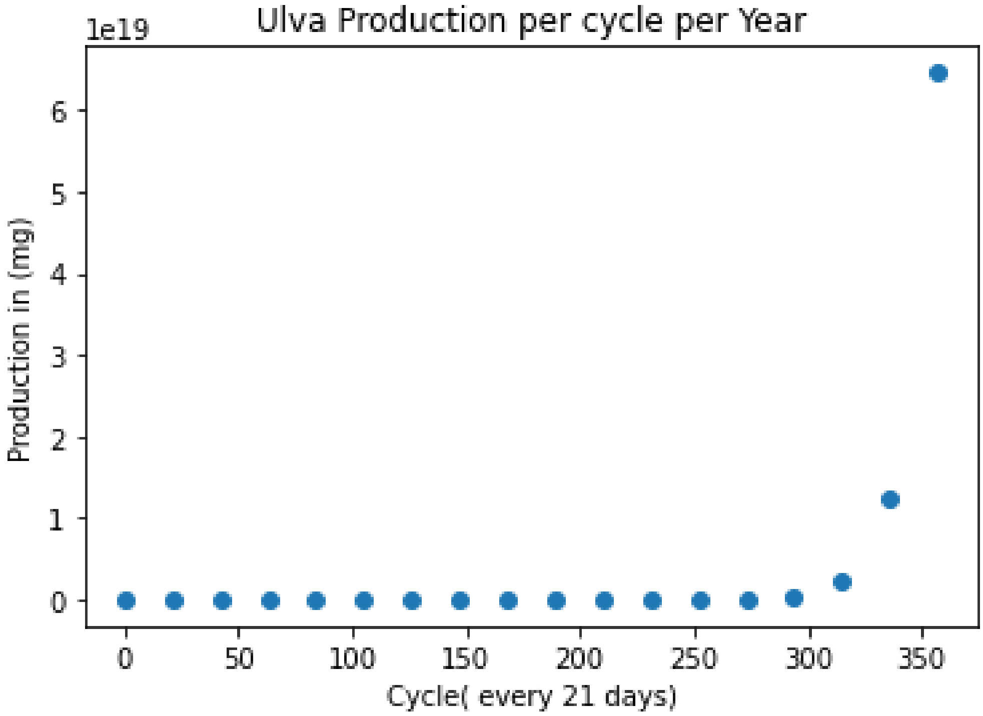
# 3.3 both with Actual concentration

```
header = [“medium”,”sample”,”day_0”,”day_7”,”day_14”,”day_21”]
data = [[‘Actual’,’growth_1’,359.2,968,1368.9,2029],
        [‘Actual’,’growth_2’,439.6,811.6,1217.5,2121],
        [‘Actual’,’growth_3’,286.8,547.5,796.6,1181.9]]

# Write the model estimated output values to a csv file (‘both_actual.csv’)
with open(‘both_actual.csv’, ‘w’, encoding=’UTF8’, newline=’’) as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

both_actual = pd.read_csv(‘both_actual.csv’)
print(“both_actual: \n”, both_actual)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for I in range(len(both_actual)):
    x1 = np.arange(0, 22, 7)
    y1 = np.array(both_actual.iloc[I, 2:6])
```

```
try:
    a = forecast(x1, y1, i)
    forecast_list.append(a)
    writeForecastIntoFile('both_actual_forecasted.csv',forecast_list )

    displayForecast('both_actual_forecasted.csv')

finally:
    print("Cannot forecast due to negative extrapolation. Poor performance")
    break
```



```
both_actual:
    medium    sample  day_0  day_7  day_14  day_21
   2  Actual  growth_1  359.2  968.0  1368.9  2029.0
   2  Actual  growth_2  439.6  811.6  1217.5  2121.0
2  Actual  growth_3  286.8  547.5   796.6  1181.9
Production at the end of the year: 1.13e+20 mg
```



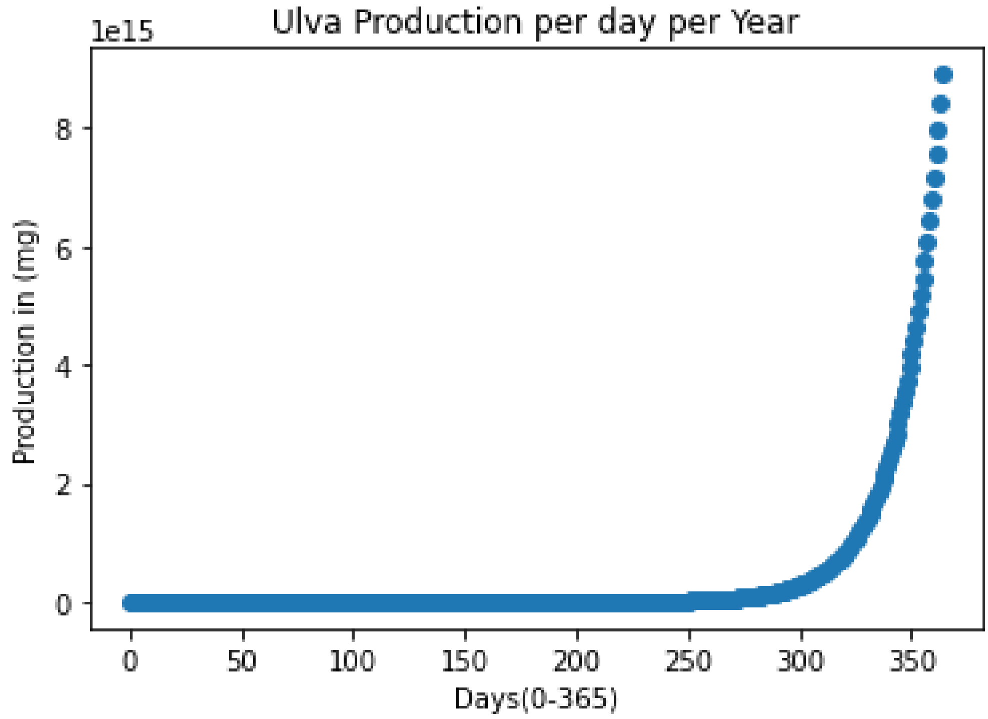
```
Cannot forecast due to negative extrapolation. Poor performance
```

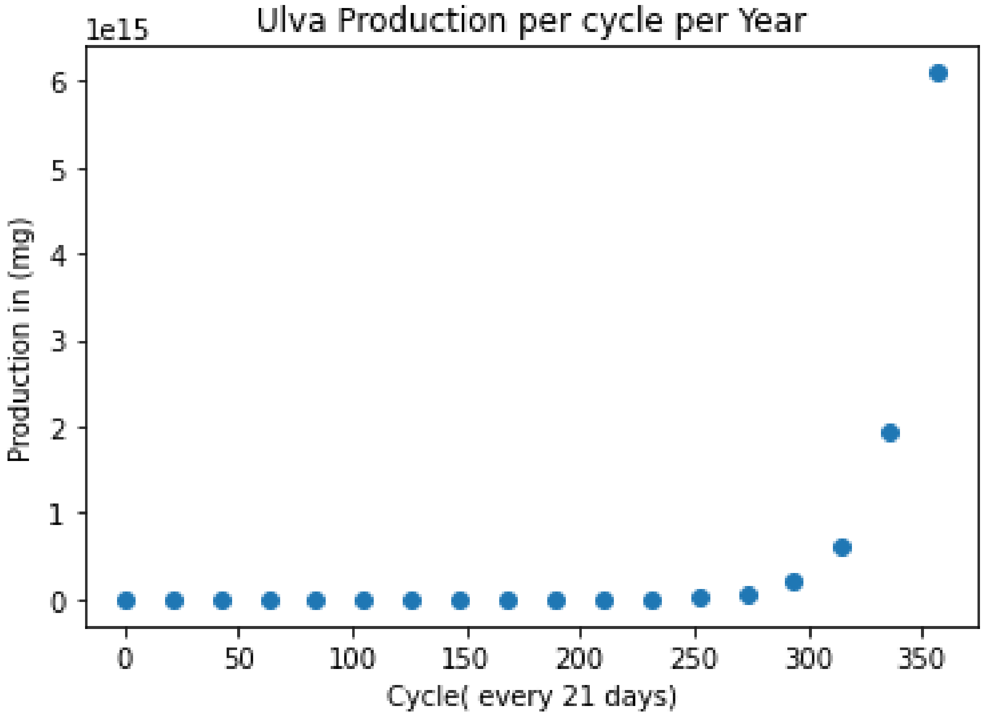
# 3.4 both with higher concentration

```
header = ["medium","sample","day_0","day_7","day_14","day_21"]
data = [['Higher','growth_1',314.7,412.7,664.9,950.4],
        ['Higher','growth_2',241.4,322.1,685.2,801.5],
        ['Higher','growth_3',257.5,278,610.3,712.7]]

# Write the model estimated output values to a csv file ('both_higher.csv')
with open('both_higher.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)

# Confirming the updated data sample

both_higher = pd.read_csv('both_higher.csv')
print("both_higher: \n", both_higher)

# Estimating the production for next upcoming days -  using Extrapolation Technique

forecast_list = []

for i in range(len(both_higher)):
    x1 = np.arange(0, 22, 7)
```

```
y1 = np.array(both_higher.iloc[i, 2:6])

try:
    a = forecast(x1, y1, i)
    forecast_list.append(a)

    writeForecastIntoFile('both_higher_forecasted.csv',forecast_list )
    displayForecast('both_higher_forecasted.csv')

finally:
    print("Cannot forecast due to negative extrapolation. Poor performance")
    break
```

```
both_higher:
    medium    sample  day_0  day_7   day_14   day_21
0  Higher  growth_1  314.7  412.7   664.9   950.4
1  Higher  growth_2  241.4  322.1   685.2   801.5
2  Higher  growth_3  257.5  278.0   610.3   712.7
Production at the end of the year: 8.91e+15 mg
```





```
Cannot forecast due to negative extrapolation. Poor performance
```

# Conclusion

Urea_Control : Production at the end of the year: 6.50e+16 mg
Urea_Lower : Production at the end of the year: 1.22e+17 mg
Urea_Actual : Production at the end of the year: 1.03e+19 mg
Urea_Higher : Production at the end of the year: 1.92e+17 mg

DAP_Control : Production at the end of the year: 6.50e+16 mg
DAP_Lower : Production at the end of the year: 1.03e+16 mg
DAP_Actual : Production at the end of the year: 1.97e+15 mg
DAP_Higher : Production at the end of the year: 6.88e+15 mg (Poor)

Both_Control : Production at the end of the year: 1.70e+25 mg (Poor)
Both_Lower : Production at the end of the year: 1.24e+18 mg
Both_Actual : Production at the end of the year: 1.13e+20 mg (Poor)
Both_Higher : Production at the end of the year: 8.91e+15 mg (Poor)