Creating NDVI Image Using sentinel data for Dharamapuri District, Tamilnadu

Installing the required Libraries

```
!pip install sentinelsat
!pip install rasterio
!pip install folium
!pip install geopandas
!pip install descartes
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
Collecting sentinelsat
  Downloading sentinelsat-1.1.1-py3-none-any.whl (48 kB)
     |████████████████████████████████| 48 kB 3.0 MB/s
Requirement already satisfied: click>=7.1 in /usr/local/lib/python3.7/dist-package
Collecting geojson>=2
  Downloading geojson-2.5.0-py2.py3-none-any.whl (14 kB)
Collecting geomet
  Downloading geomet-1.0.0-py3-none-any.whl (28 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tqdm>=4.58 in /usr/local/lib/python3.7/dist-package
Collecting html2text
  Downloading html2text-2020.1.16-py3-none-any.whl (32 kB)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loc
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist
Installing collected packages: html2text, geomet, geojson, sentinelsat
Successfully installed geojson-2.5.0 geomet-1.0.0 html2text-2020.1.16 sentinelsat-
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
Collecting rasterio
  Downloading rasterio-1.2.10-cp37-cp37m-manylinux1_x86_64.whl (19.3 MB)
     |████████████████████████████████| 19.3 MB 1.5 MB/s
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (fr
Collecting snuggs>=1.4.1
  Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (
Collecting affine
  Downloading affine-2.3.1-py2.py3-none-any.whl (16 kB)
Collecting click-plugins
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.7/dist-p
Installing collected packages: snuggs, cligj, click-plugins, affine, rasterio
Successfully installed affine-2.3.1 click-plugins-1.1.1 cligj-0.7.2 rasterio-1.2.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.7/dist-packages (0
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: jinja2>=2.9 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loc
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
Collecting geopandas
  Downloading geopandas-0.10.2-py2.py3-none-any.whl (1.0 MB)
     |████████████████████████████████| 1.0 MB 4.8 MB/s
Requirement already satisfied: shapely>=1.6 in /usr/local/lib/python3.7/dist-packa
```

```
import folium
import os
import numpy as np


from sentinelsat import SentinelAPI
import geopandas as gpd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from shapely.geometry import MultiPolygon, Polygon
import rasterio as rio
from rasterio.plot import show
import rasterio.mask
```

Loging in to the copernicus data hub

```
user = 'ssgis'
password = 'saravanan@3045'

api = SentinelAPI(user, password, 'https://scihub.copernicus.eu/dhus')
```

Reading the Shapefile Geopandas and visualize it with Folium python library.

```
!wget https://www.dropbox.com/s/g24n4lme4722mfo/data.zip
```

```
--2022-11-15 10:09:53--  https://www.dropbox.com/s/g24n4lme4722mfo/data.zip
Resolving www.dropbox.com (www.dropbox.com)... 162.125.5.18, 2620:100:601d:18::a27d:5
Connecting to www.dropbox.com (www.dropbox.com)|162.125.5.18|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: /s/raw/g24n4lme4722mfo/data.zip [following]
```

```
--2022-11-15 10:09:53--  https://www.dropbox.com/s/raw/g24n4lme4722mfo/data.zip
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://uc60a82c09e57054555dbeac9261.dl.dropboxusercontent.com/cd/0/inline/
--2022-11-15 10:09:54--  https://uc60a82c09e57054555dbeac9261.dl.dropboxusercontent.c
Resolving uc60a82c09e57054555dbeac9261.dl.dropboxusercontent.com (uc60a82c09e5705455
Connecting to uc60a82c09e57054555dbeac9261.dl.dropboxusercontent.com (uc60a82c09e570
HTTP request sent, awaiting response... 302 Found
Location: /cd/0/inline2/Bwwv0Wf5XCdny8YQ_OSDDzK1-vlYliU6wblnibLFuC0I3OkMfSZcBPRMye0h
--2022-11-15 10:09:54--  https://uc60a82c09e57054555dbeac9261.dl.dropboxusercontent.c
Reusing existing connection to uc60a82c09e57054555dbeac9261.dl.dropboxusercontent.com
HTTP request sent, awaiting response... 200 OK
Length: 29371 (29K) [application/zip]
Saving to: ‘data.zip’

data.zip            100%[===================>]  28.68K  --.-KB/s    in 0.01s

2022-11-15 10:09:54 (1.96 MB/s) - ‘data.zip’ saved [29371/29371]
```

```
!unzip 'data.zip'
```

```
  inflating: data/.git/hooks/applypatch-msg.sample
  inflating: data/.git/hooks/commit-msg.sample
  inflating: data/.git/hooks/fsmonitor-watchman.sample
  inflating: data/.git/hooks/post-update.sample
  inflating: data/.git/hooks/pre-applypatch.sample
  inflating: data/.git/hooks/pre-commit.sample
  inflating: data/.git/hooks/pre-merge-commit.sample
  inflating: data/.git/hooks/pre-push.sample
  inflating: data/.git/hooks/pre-rebase.sample
  inflating: data/.git/hooks/pre-receive.sample
  inflating: data/.git/hooks/prepare-commit-msg.sample
  inflating: data/.git/hooks/push-to-checkout.sample
  inflating: data/.git/hooks/update.sample
  inflating: data/.git/index
   creating: data/.git/info/
  inflating: data/.git/info/exclude
   creating: data/.git/logs/
  inflating: data/.git/logs/HEAD
   creating: data/.git/logs/refs/
   creating: data/.git/logs/refs/heads/
  inflating: data/.git/logs/refs/heads/main
   creating: data/.git/logs/refs/remotes/
   creating: data/.git/logs/refs/remotes/origin/
  inflating: data/.git/logs/refs/remotes/origin/main
   creating: data/.git/objects/
   creating: data/.git/objects/4b/
  inflating: data/.git/objects/4b/825dc642cb6eb9a060e54bf8d69288fbee4904
   creating: data/.git/objects/61/
  inflating: data/.git/objects/61/0eb6e2c691f9392550aafcad4d5ba0b44a2894
   creating: data/.git/objects/74/
  inflating: data/.git/objects/74/2e45a39115a6b56a367af5768b0cbf16dbf037
   creating: data/.git/objects/ea/
  inflating: data/.git/objects/ea/12c2d458693b6959d64404d79e2cec131cc4d4
   creating: data/.git/objects/info/
   creating: data/.git/objects/pack/
   creating: data/.git/refs/
   creating: data/.git/refs/heads/
```

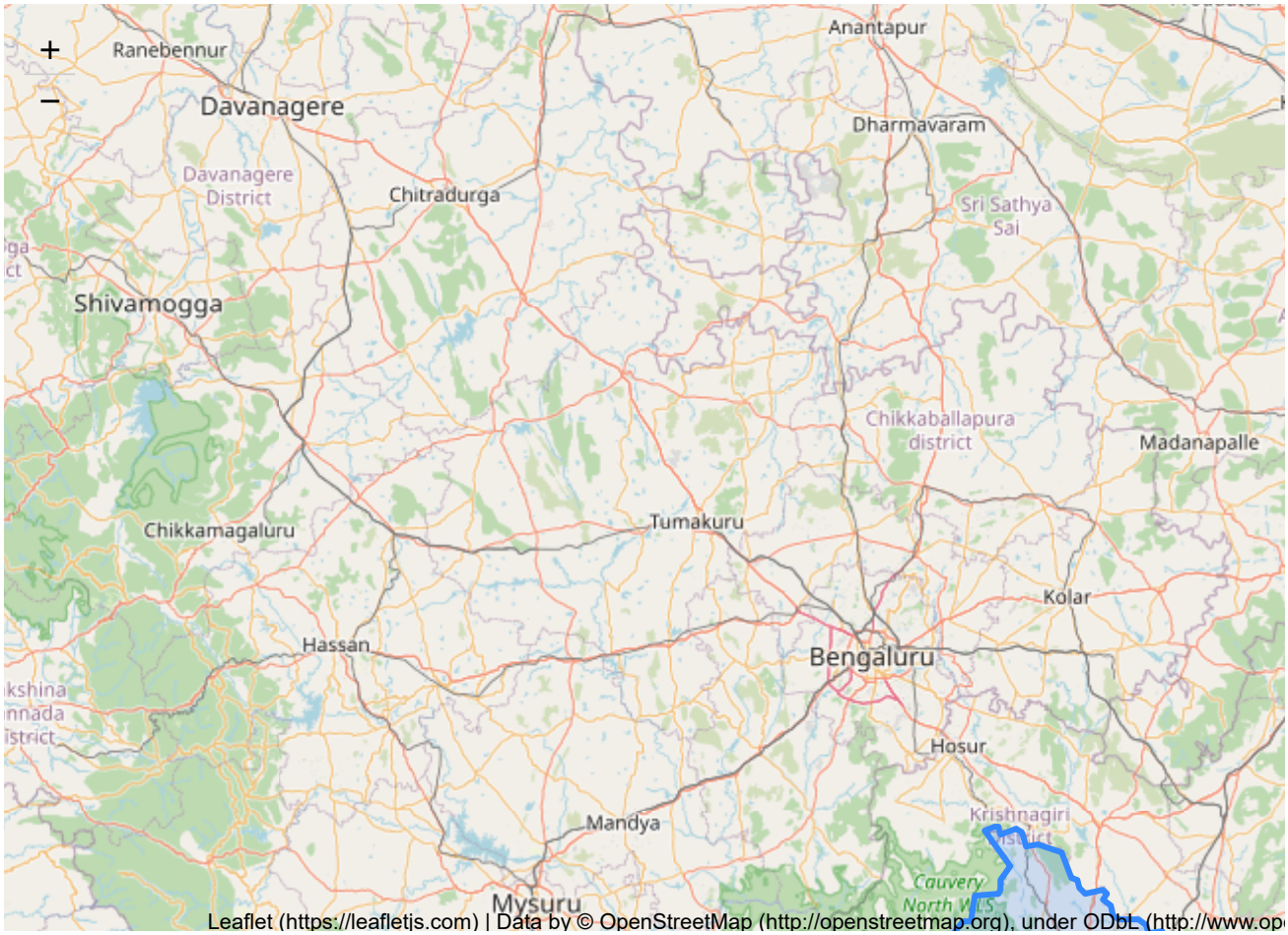
```
   creating: data/.git/refs/heads/
  inflating: data/.git/refs/heads/main
   creating: data/.git/refs/remotes/
   creating: data/.git/refs/remotes/origin/
  inflating: data/.git/refs/remotes/origin/main
   creating: data/.git/refs/tags/
   creating: data/Dharmapuri/
  inflating: data/Dharmapuri/Dharmapuri.cpg
  inflating: data/Dharmapuri/Dharmapuri.dbf
  inflating: data/Dharmapuri/Dharmapuri.prj
  inflating: data/Dharmapuri/Dharmapuri.sbn
  inflating: data/Dharmapuri/Dharmapuri.sbx
  inflating: data/Dharmapuri/Dharmapuri.shp
  inflating: data/Dharmapuri/Dharmapuri.shx
  inflating: data/Dharmapuri/overlay.cpg
  inflating: data/Dharmapuri/overlay.dbf
  inflating: data/Dharmapuri/overlay.prj
  inflating: data/Dharmapuri/overlay.sbn
  inflating: data/Dharmapuri/overlay.sbx
  inflating: data/Dharmapuri/overlay.shp
  inflating: data/Dharmapuri/overlay.shx
  inflating: data/README.md
```

```
boundary = gpd.read_file('data/Dharmapuri/Dharmapuri.shp')
boundary
```

	cartodb_id	censuscode	dt_cen_cd	st_cen_cd	st_nm	district	
0	148	630.0	29	33	Tamil Nadu	Dharmapuri	POLYGON (

```
map = folium.Map([12.11, 78.2091], zoom_start=9)

folium.GeoJson(boundary).add_to(map)
map
```



With the above code, the boundary shapefile is read using Geopandas and called as boundary, then later created an empty base map in Folium centred around coordinates in the area, and defined it as map. Finally, The Geopanda data has been added to the base mapand we can visualize it.

Creating the footprint

```
bound = gpd.read_file('data/Dharmapuri/overlay.shp')

footprint = None
for i in bound['geometry']:
    footprint = i
```

Created a query for Sentinel 2 images Level 2A with cloud coverage between 0 and 10 that fall or intersect with the footprint (Area of study). For the time period, we are interested in Sentinel Level 2A satellite images taken between '20220501' and '20220530'

```
products = api.query(footprint,
                     date = ('20220501', '20220530'),
                     platformname = 'Sentinel-2',
                     processinglevel = 'Level-2A',
                     cloudcoverpercentage = (0,10))
```

From here we can create a GeodataFrame or Dataframe from the product dictionary and sort them according to cloud coverage percentage.

```
products_gdf = api.to_geodataframe(products)
products_gdf_sorted = products_gdf.sort_values(['cloudcoverpercentage'], ascending=[True])
products_gdf_sorted
```

	title	
053a906a-6f8d-4769-9a20-47432d197d96	S2B_MSIL2A_20220510T100029_N0400_R122_T33TTG_2...	https://scihub.copernic
0acc4d67-e0e4-4736-ae29-73c94dc0e7ee	S2B_MSIL2A_20220510T100029_N0400_R122_T32TQM_2...	https://scihub.copernic

2 rows × 41 columns

Let us say we are interested in the first satellite image since this has the least cloud coverage of all available images. we can simply call download and provide the product name

```
api.download("053a906a-6f8d-4769-9a20-47432d197d96")
```

```
Downloading
S2B_MSIL2A_20220510T100029_N0400_R122_T33TTG_20220510T122841.zip:
100%
{'id': '053a906a-6f8d-4769-9a20-47432d197d96',
 'title': 'S2B_MSIL2A_20220510T100029_N0400_R122_T33TTG_20220510T122841',
 'size': 1006521085,
 'md5': 'c4272fe74be83fa9a9e98b04f6e65019',
 'date': datetime.datetime(2022, 5, 10, 10, 0, 29, 24000),
 'footprint': 'POLYGON((11.354954233772318 42.39470139413761,12.687592758603905
42.42936087235922,12.723033432665233 41.441212166985935,11.410731332148327
41.407726190110715,11.354954233772318 42.39470139413761))',
 'url': "https://scihub.copernicus.eu/dhus/odata/v1/Products('053a906a-6f8d-4769-9a2(
47432d197d96')/$value",
 'Online': True,
 'Creation Date': datetime.datetime(2022, 5, 10, 16, 29, 9, 665000),
```

```
!wget https://www.dropbox.com/s/hxilp617uvlbot6/S2A_MSIL1C_20220528T050701_N0400_R019_T43P
```

```
!unzip 'S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315.zip'
```

```
Archive:  S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315.zip
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
   creating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
  inflating: S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL
```

Create RGB Image

```
source ='S2A_MSIL1C_20220528T050701_N0400_R019_T43PHP_20220528T070315/S2A_MSIL1C_20220528T
```

```
# Open Bands 4, 3 and 2 with Rasterio
b4 = rio.open(source+'/T43PHP_20220528T050701_B04.jp2')
b3 = rio.open(source+'/T43PHP_20220528T050701_B03.jp2')
b2 = rio.open(source+'/T43PHP_20220528T050701_B02.jp2')
```

```
with rio.open('RGB_Image.tiff','w',driver='Gtiff', width=b4.width, height=b4.height,
              count=3,crs=b4.crs,transform=b4.transform, dtype=b4.dtypes[0]) as rgb:
    rgb.write(b2.read(1),1)
    rgb.write(b3.read(1),2)
    rgb.write(b4.read(1),3)
    rgb.close()
```

```
raster = rio.open("RGB_Image.tiff")
```

NDVI

To calculate the NDVI, we need Red band and Near-Infrared Band (NIR).Sentinel Images have red in 4th band and NIR in the 8th band.

The formula for NDVI calculation is: nir - red /(nir + red).

First we need to read the 4th and 8th bands as arrays and creating NDVI Index using the above formula.

```
b4 = rio.open(source+'/T43PHP_20220528T050701_B04.jp2')
b8 = rio.open(source+'/T43PHP_20220528T050701_B08.jp2')
```

```
red = b4.read()
nir = b8.read()
```

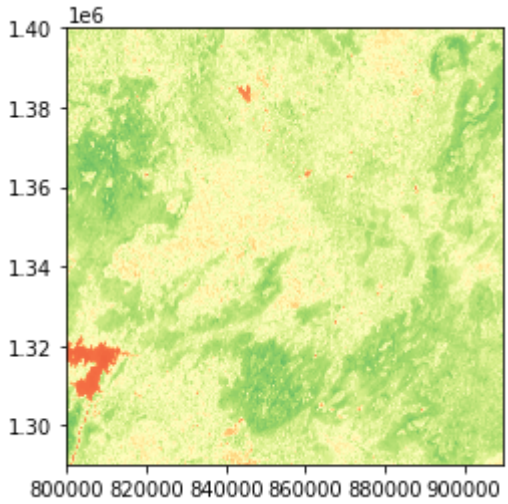
```
ndvi = (nir.astype(float)-red.astype(float))/(nir+red)
```

```
meta = b4.meta

meta.update(driver='GTiff')
meta.update(dtype=rasterio.float32)
```

```
with rasterio.open('NDVI2.tif', 'w', **meta) as dst:
    dst.write(ndvi.astype(rasterio.float32))


ndvi = rio.open(r'NDVI2.tif')
show(ndvi, transform=ndvi.transform, cmap='RdYlGn')
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9281057ed0>
```

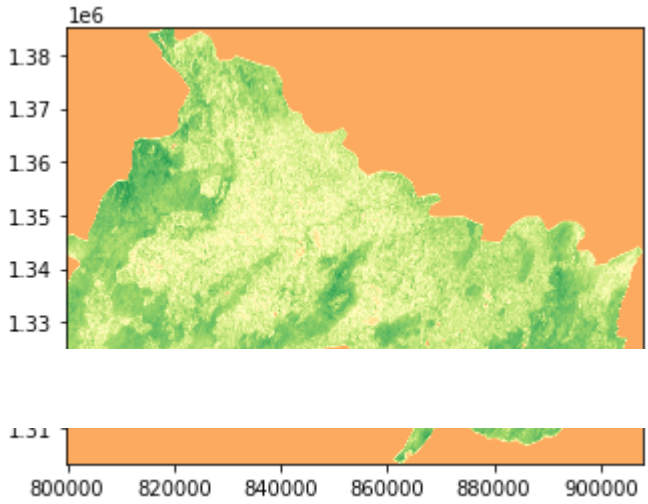
Mask Satellite image

```
boundary_proj = boundary.to_crs({'init': 'epsg:32643'})

with rio.open("/content/NDVI2.tif") as src:
    out_image, out_transform = rio.mask.mask(src, boundary_proj.geometry,crop=True)
    out_meta = src.meta.copy()
    out_meta.update({"driver": "GTiff",
                "height": out_image.shape[1],
                "width": out_image.shape[2],
                "transform": out_transform})

with rio.open("NDVI_masked.tif", "w", **out_meta) as dest:
    dest.write(out_image)


ndvi_mask = rio.open(r'NDVI_masked.tif')
show(ndvi_mask, transform=ndvi_mask.transform, cmap='RdYlGn')
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7f927f2c32d0>
```

Colab paid products - Cancel contracts here

✓ 0s completed at 3:54 PM