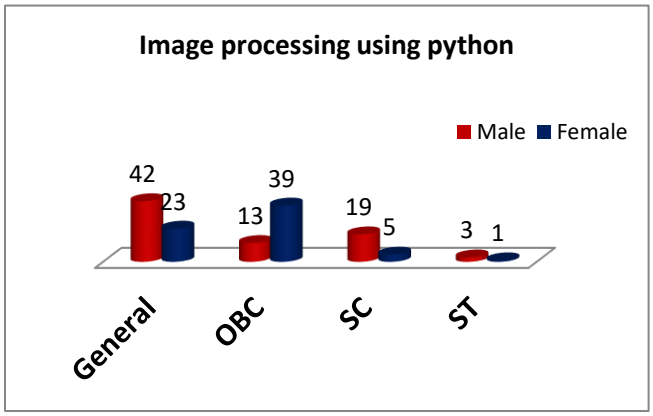
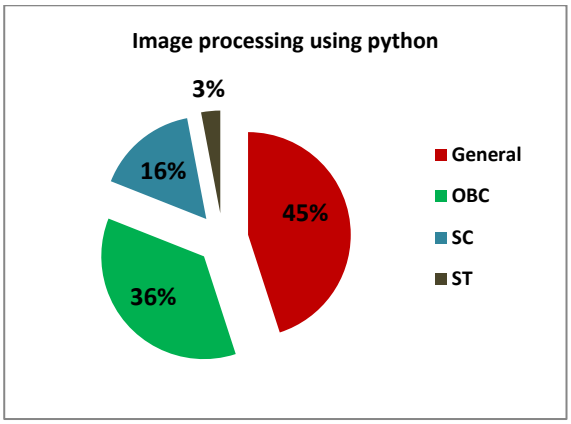
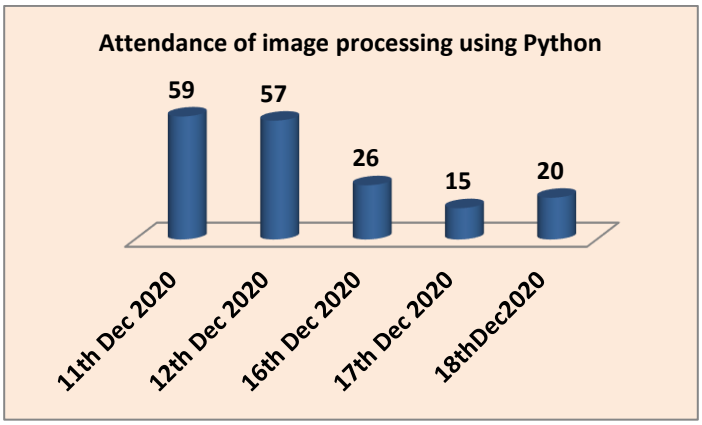
Training – Image Processing using Python		
Target Group – Professors, Scientists, RA, SF, YP-II, PG and Ph.D. Students		
Date of Training	11th Jan – 15th Jan, 2021	
Time	3.30 p.m. – 5.00 p.m.	
Location	CAE, JNKVV, JABALPUR	
1. Training Objective		
Image Processing using Python Training Program under NAHEP-CAAST-CSDA Project.		
2. Participants		
Coordinator - Dr. M.K. Awasthi, Co-PI, Dr. Sourabh Nema, Dr. Minakshi Meshram, Dr. Popat Shivaji Pawar, Dr. Umakant Rawat, Dr. Devendra Vasht, Pratiman Patel, Aniket Rajput, Anjali Patel, Rachit Nema, Krishna Singh, Pratima Pathak.		
3. Contents of event		
Date	**Speaker**	**Course Topic**
8th Dec 2020 (Day 1)	Mr. Sathish Singh	Satellites image vs. normal image
9th Dec 2020 (Day 2)	Mr. Sathish Singh	Libraries specific to satellite image
10th Dec 2020 (Day 3)	Mr. Sathish Singh	Introduction to rasterio
11th Dec 2020 (Day 4)	Mr. Sathish Singh	Satellite image processing using rasterio
12thDec2020 (Day 5)	Mr. Sathish Singh	Rasterio, rasterstates, geopandas, geoplots, Fiona, matplotlip, pandas, numpy
Discussion with participants	Image processing has been used to create weird and beautiful modification to pictures many of us have seen online. Older black and white photos can be brought to life using colorization techniques. Color photos can be made to look like old black and white photos. We will create python application that reads of a list of images, modifies their size and appearance and saves the images in another directory.	
Impact of training	One hundred forty five participants registered for the course and out of these fifty nine students have undergone attend the training on the subject. Total male participants were 77 and female participants 68. Percentage of participants in different category was UR- 45%, SC- 16%, ST- 3% and OBC- 36%.The image processing is too big to comprehend. It has been backbone of many industry including Deep learning. It is used across multiple places.	

Gender & Category wise Distribution of Participants-

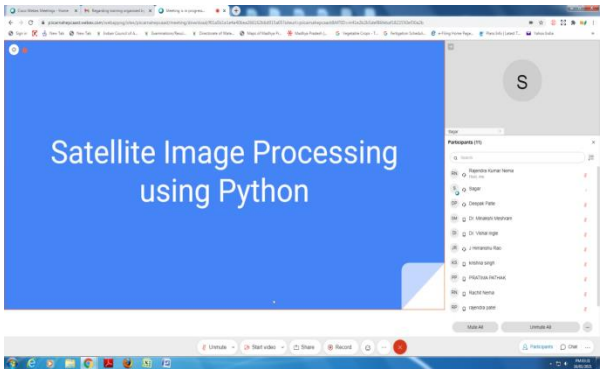
Number of registered participants						% of participants in diff. category			
Category	General	OBC	SC	ST	Total	General	OBC	SC	ST
Male	42	13	19	03	77	55	17	25	3
Female	23	39	05	01	68	34	57	8	1
Total	65	52	24	04	145	45	36	16	3

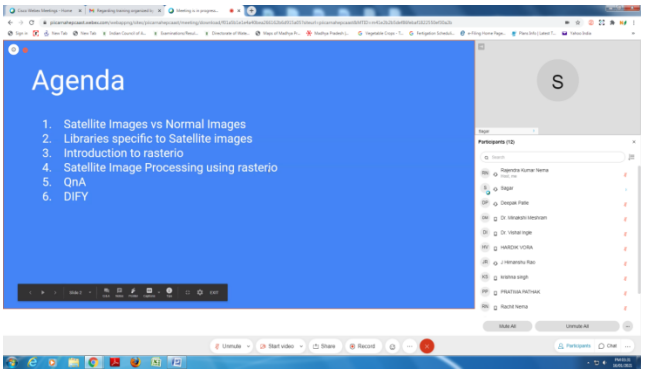


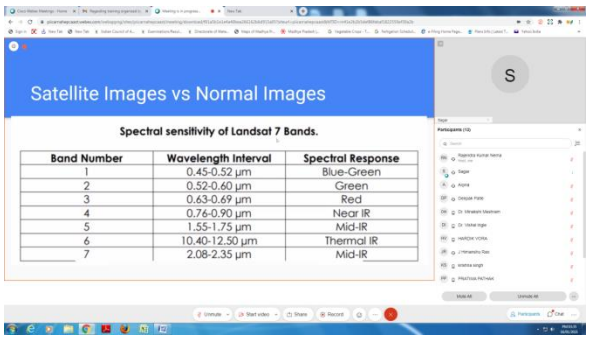


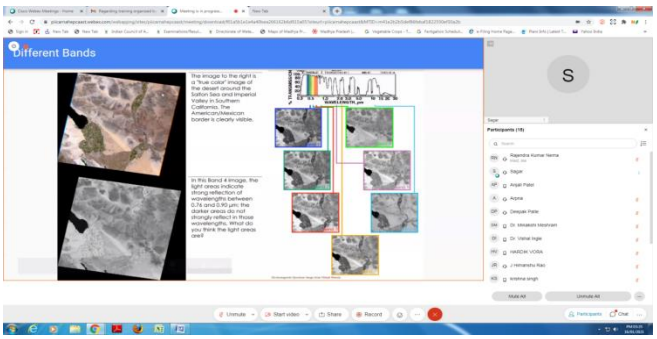


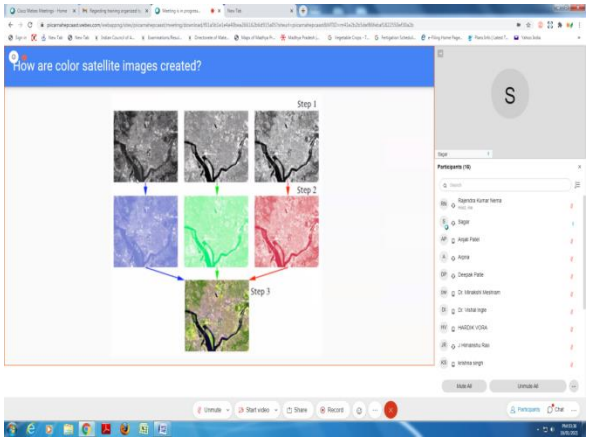
Date	Attendance
11th Dec 2020	59
12th Dec 2020	57
16th Dec 2020	26
17th Dec 2020	15
18thDec2020	20

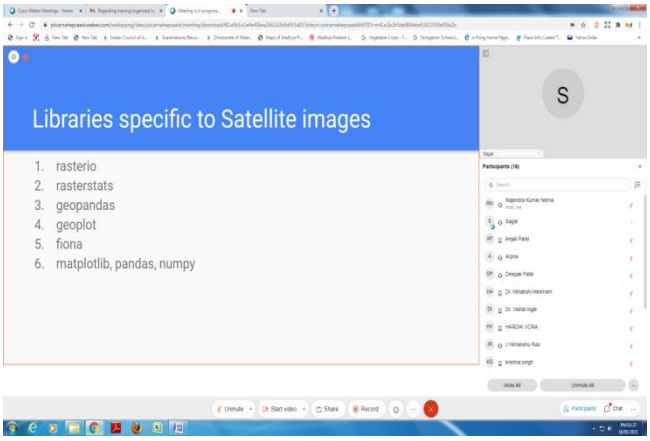

Satellite Image Processing
using Python

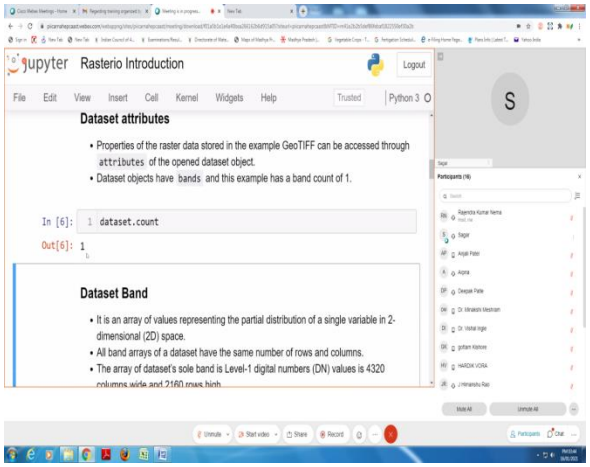

Agenda
1. Satellite Images vs Normal Images
2. Libraries specific to Satellite images
3. Introduction to rasterio
4. Satellite Image Processing using rasterio
5. QnA
6. DIFY

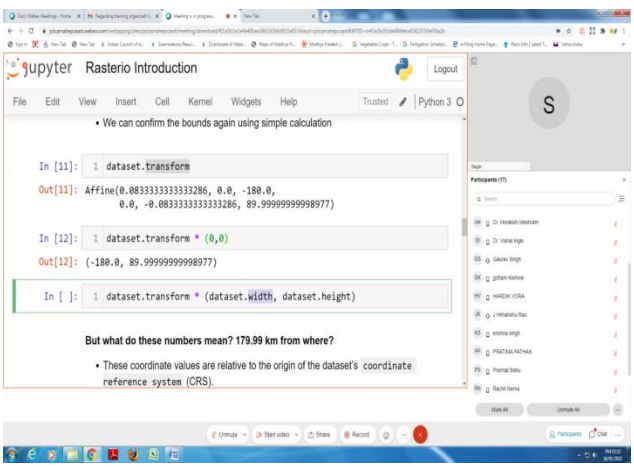

Satellite Images vs Normal Images
Spectral sensitivity of Landsat 7 Bands.
Band Number | Wavelength Interval | Spectral Response
1 | 0.45-0.52 µm | Blue-Green
2 | 0.52-0.60 µm | Green
3 | 0.63-0.69 µm | Red
4 | 0.76-0.90 µm | Near IR
5 | 1.55-1.75 µm | Mid-IR
6 | 10.40-12.50 µm | Thermal IR
7 | 2.08-2.35 µm | Mid-IR

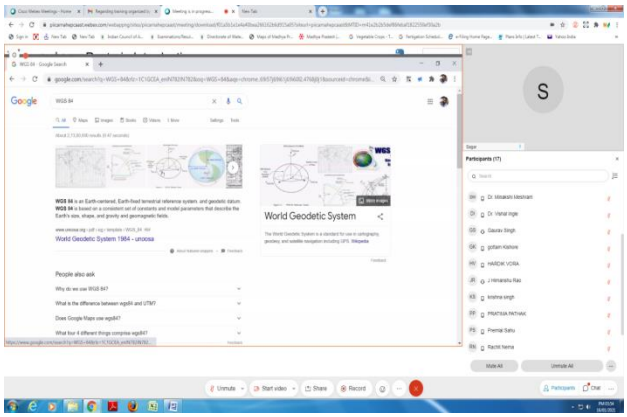

Different Bands

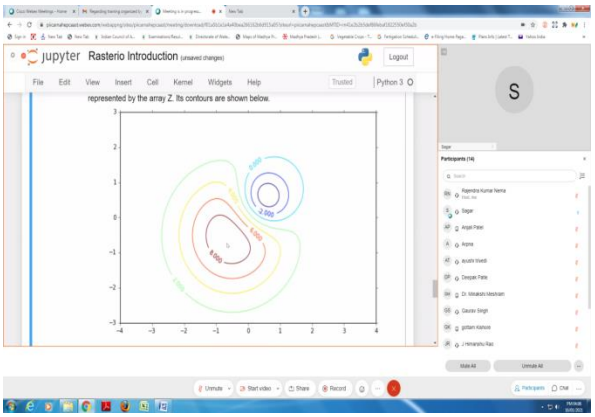

How are color satellite images created?
Step 1
Step 2
Step 3

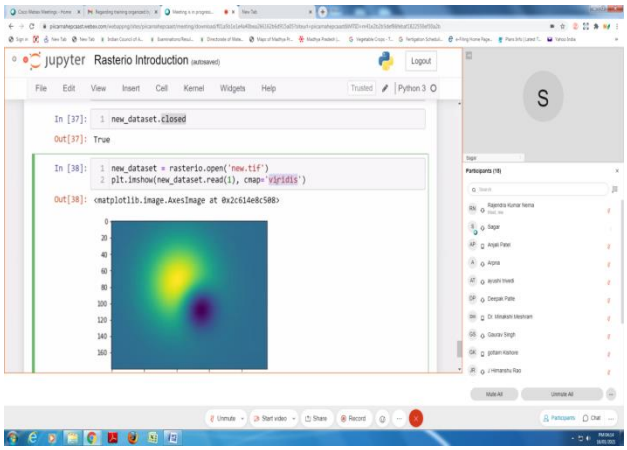

Libraries specific to Satellite images
1. rasterio
2. rasterstats
3. geopandas
4. geoplot
5. fiona
6. matplotlib, pandas, numpy

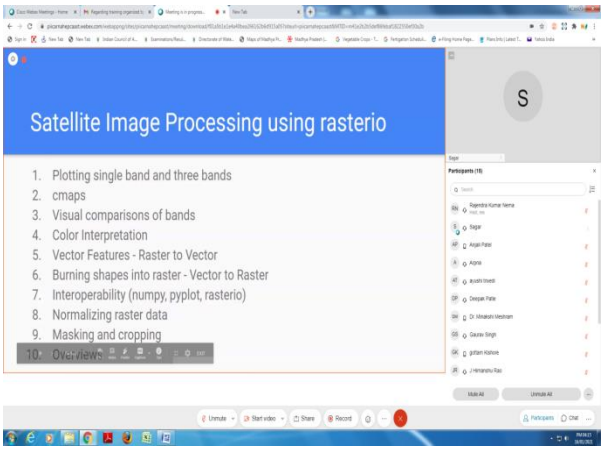

Jupyter Rasterio Introduction
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Dataset attributes
• Properties of the raster data stored in the example GeoTIFF can be accessed through attributes of the opened dataset object.
• Dataset objects have bands and this example has a band count of 1.
In [6]: 1 dataset.count
Out[6]: 1
Dataset Band
• It is an array of values representing the partial distribution of a single variable in 2-dimensional (2D) space.
• All band arrays of a dataset have the same number of rows and columns.
• The array of dataset's sole band is Level-1 digital numbers (DN) values is 4320

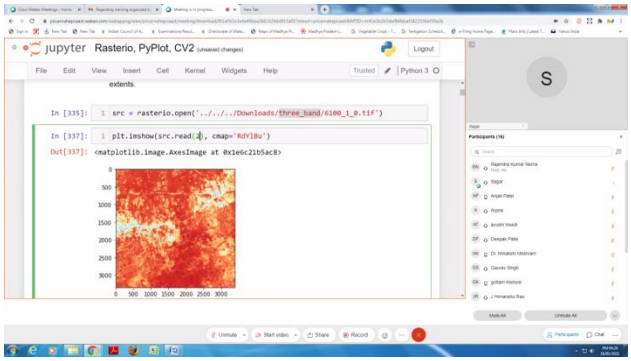

Jupyter Rasterio Introduction
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
• We can confirm the bounds again using simple calculation
In [11]: 1 dataset.transform
Out[11]: Affine(0.0833333333333286, 0.0, -180.0,
0.0, -0.0833333333333286, 89.99999999998977)
In [12]: 1 dataset.transform * (0,0)
Out[12]: (-180.0, 89.99999999998977)
In [ ]: 1 dataset.transform * (dataset.width, dataset.height)
But what do these numbers mean? 179.99 km from where?
• These coordinate values are relative to the origin of the dataset's coordinate reference system (CRS).

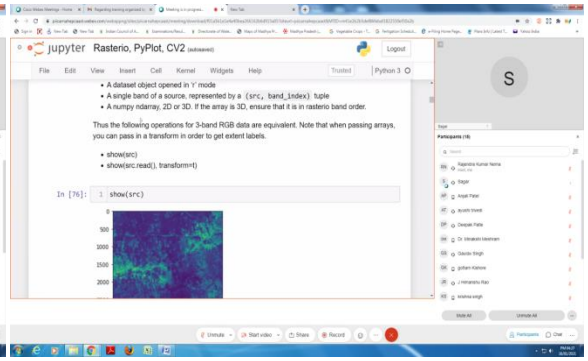

World Geodetic System

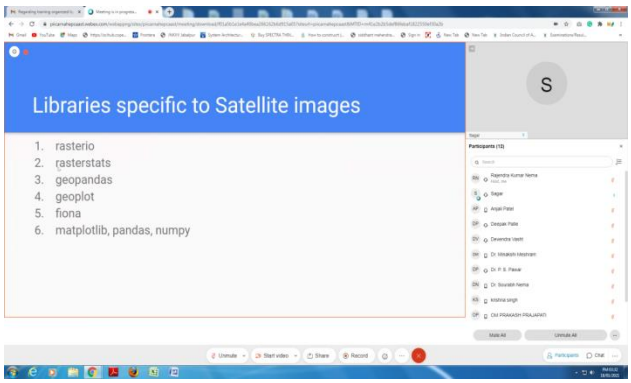

jupyter Rasterio Introduction (unsaved changes)
represented by the array Z. Its contours are shown below.

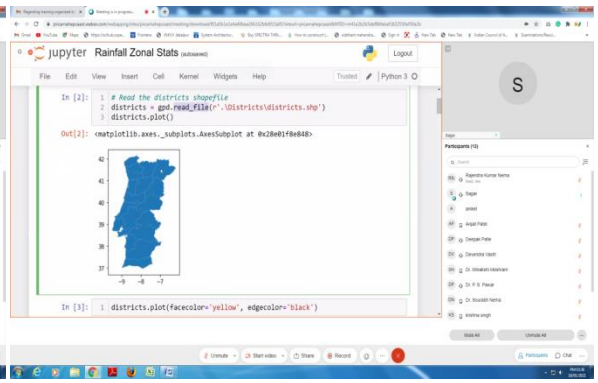

jupyter Rasterio Introduction (autosaved)
In [37]: new_dataset.closed
Out[37]: True
In [38]: new_dataset = rasterio.open('new.tif')
plt.imshow(new_dataset.read(1), cmap='viridis')
Out[38]: <matplotlib.image.AxesImage at 0x2c614e8c508>

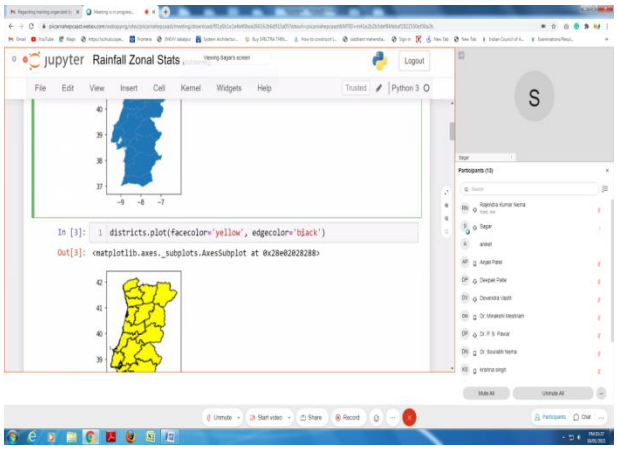

Satellite Image Processing using rasterio
1. Plotting single band and three bands
2. cmaps
3. Visual comparisons of bands
4. Color Interpretation
5. Vector Features - Raster to Vector
6. Burning shapes into raster - Vector to Raster
7. Interoperability (numpy, pyplot, rasterio)
8. Normalizing raster data
9. Masking and cropping
10. Overviews

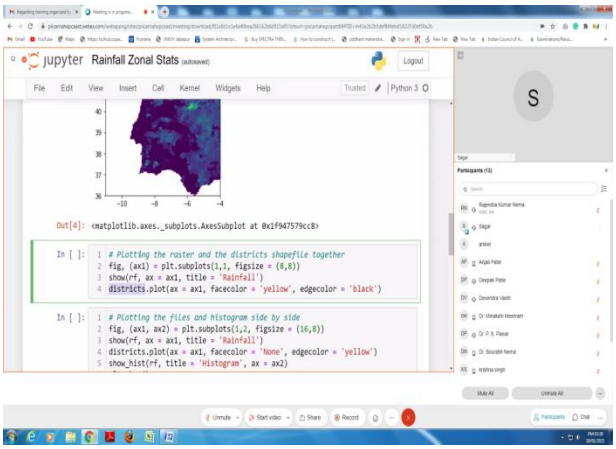

jupyter Rasterio, PyPlot, CV2 (unsaved changes)
In [335]: src = rasterio.open('../../../Downloads/three_band/6100_1_0.tif')
In [337]: plt.imshow(src.read(2), cmap='RdYlBu')
Out[337]: <matplotlib.image.AxesImage at 0x1e6c21b5ac8>

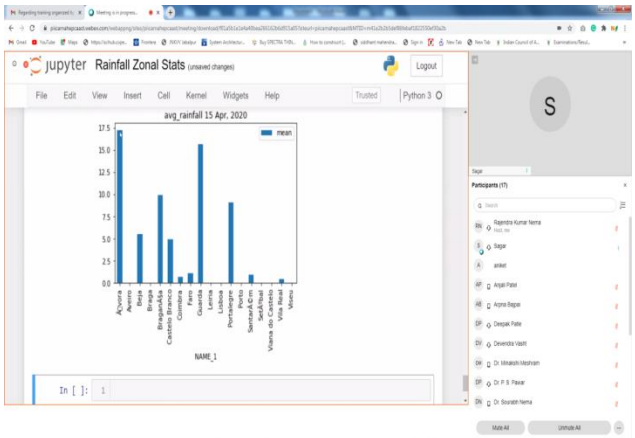

jupyter Rasterio, PyPlot, CV2 (autosaved)
• A dataset object opened in 'r' mode
• A single band of a source, represented by a (src, band_index) tuple
• A numpy ndarray, 2D or 3D. If the array is 3D, ensure that it is in rasterio band order.
Thus the following operations for 3-band RGB data are equivalent. Note that when passing arrays, you can pass in a transform in order to get extent labels.
• show(src)
• show(src.read(), transform=t)
In [76]: show(src)

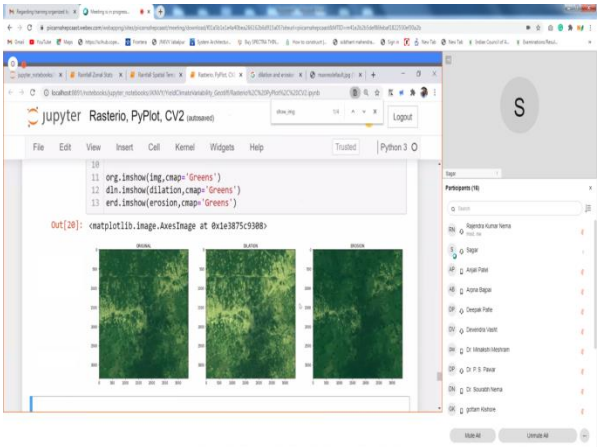

Libraries specific to Satellite images
1. rasterio
2. rasterstats
3. geopandas
4. geoplot
5. fiona
6. matplotlib, pandas, numpy

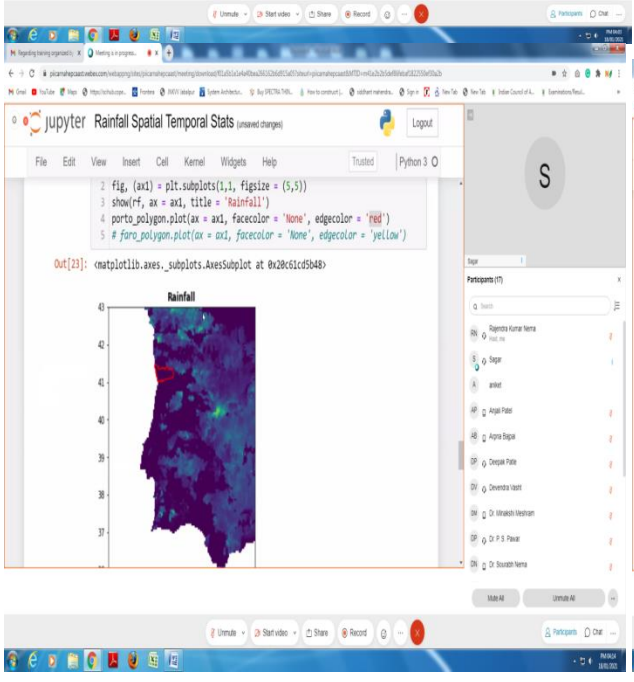

jupyter Rainfall Zonal Stats (autosaved)
In [2]: # Read the districts shapefile
districts = gpd.read_file(r'.\Districts\districts.shp')
districts.plot()
Out[2]: <matplotlib.axes._subplots.AxesSubplot at 0x28e01f8e848>
In [3]: districts.plot(facecolor='yellow', edgecolor='black')

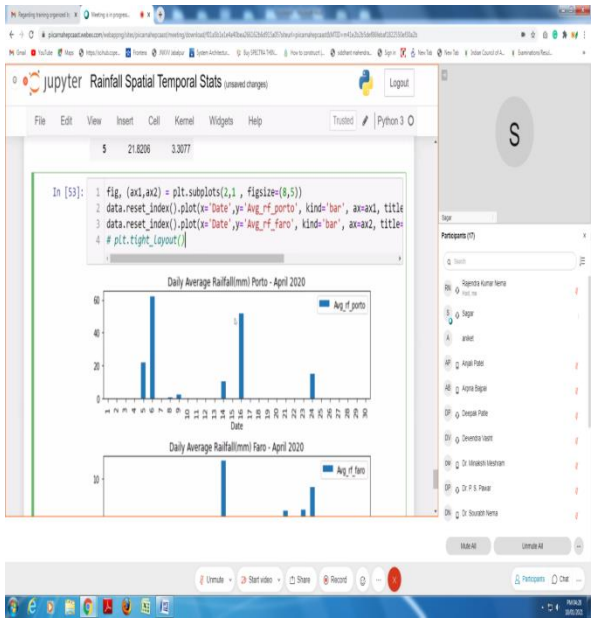

jupyter Rainfall Zonal Stats
In [3]: districts.plot(facecolor='yellow', edgecolor='black')
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x28e02028288>


jupyter Rainfall Zonal Stats (autosaved)
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1f947579cc8>
In [ ]: # Plotting the raster and the districts shapefile together
fig, (ax1) = plt.subplots(1,1, figsize = (8,8))
show(rf, ax = ax1, title = 'Rainfall')
districts.plot(ax = ax1, facecolor = 'yellow', edgecolor = 'black')
In [ ]: # Plotting the files and histogram side by side
fig, (ax1, ax2) = plt.subplots(1,2, figsize = (16,8))
show(rf, ax = ax1, title = 'Rainfall')
districts.plot(ax = ax1, facecolor = 'None', edgecolor = 'yellow')
show_hist(rf, title = 'Histogram', ax = ax2)


jupyter Rainfall Zonal Stats (unsaved changes)
avg_rainfall 15 Apr, 2020
mean
NAME_1
In [ ]: 1


jupyter Rasterio, PyPlot, CV2 (autosaved)
org.imshow(img,cmap='Greens')
dln.imshow(dilation,cmap='Greens')
ero.imshow(erosion,cmap='Greens')
Out[20]: <matplotlib.image.AxesImage at 0x1e38775c9388>


jupyter Rainfall Spatial Temporal Stats (unsaved changes)
fig, (ax1) = plt.subplots(1,1, figsize = (5,5))
show(rf, ax = ax1, title = 'Rainfall')
porto_polygon.plot(ax = ax1, facecolor = 'None', edgecolor = 'red')
# faro_polygon.plot(ax = ax1, facecolor = 'None', edgecolor = 'yellow')
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x20c61cd5b48>
Rainfall


jupyter Rainfall Spatial Temporal Stats (unsaved changes)
5 21.8206 3.3077
In [53]: fig, (ax1,ax2) = plt.subplots(2,1 , figsize=(8,5))
data.reset_index().plot(x='Date',y='Avg_rf_porto', kind='bar', ax=ax1, title
data.reset_index().plot(x='Date',y='Avg_rf_faro', kind='bar', ax=ax2, title=
plt.tight_layout()
Daily Average Raifall(mm) Porto - April 2020
Avg_rf_porto
Date
Daily Average Raifall(mm) Faro - April 2020
Avg_rf_faro