Project: Change Detection Through Remote Sensing

Task 1: Explore different methods used for upscaling of a image

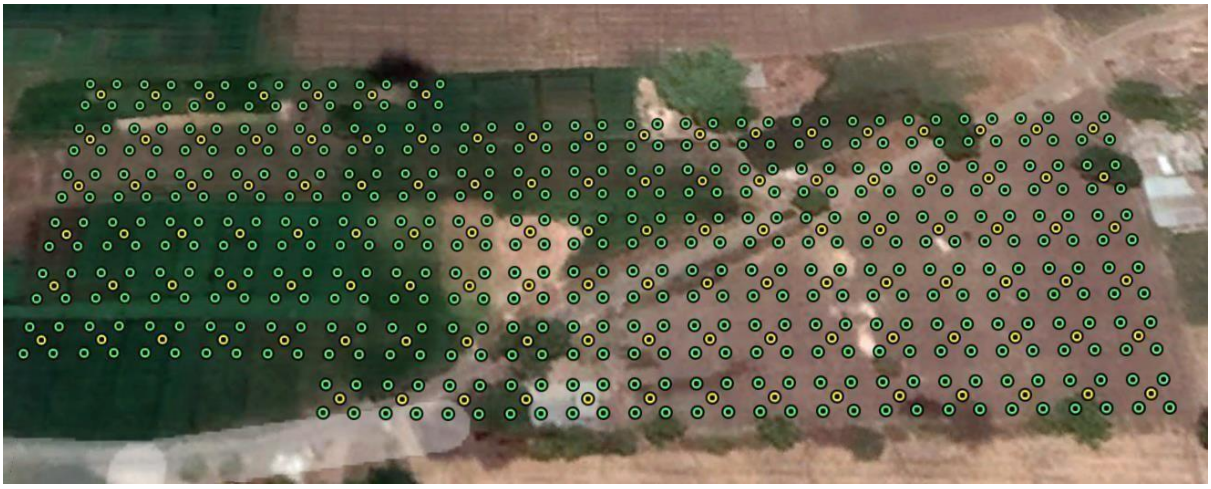
My findings: I found 3 methods of upscaling the resolution of a image.

1. Nearest Neighbour Method
2. Bilinear Method
3. Bicubic Method

According to me bilinear method was the best method which suits to this problem statement as it predicts the missing pixel value by adjusting the value of the adjacent known pixels according to how far they are from the missing pixel.

Task 2: Breaking a point with a particular longitude and latitude into a 4 different points

My findings: There is a general formula which gives the new latitude and longitude by adding some distance to the original latitude and longitude. Applying that formula on the given longitudes and latitude we get 2 new each lat and long intersecting them with each other we get 4 new coordinates for each 1 coordinate. That's exactly what was needed.



Yellow dots- original point

Green dots- new points

Task 3: Explore and Understand the concept of NDVI and working of satellite "Sentinal-2"

My findings: I found that sentinal2 is satellite which rotates around the earth and sends rays with different wavelength. And these rays are classified in 13 bands with 3 bands which have 1 box of 60 by 60 meter, 6 bands with range of 20 by 20 meter and 4 bands with range of 10 by 10 meters. Now the problem lies here that the rays are sent in smallest of 10m square and

everything in that 10 m is cumulatively a uniform value so we need to break that 10 m block into 4 of 5m each.

NDVI- Normalized Difference Vegetation Index

NDVI is a index derived cy these combination of bands to represent the vegetation in a particular block this is a gray scale band where the value of this index lies from -1 to 1

Formula of NDVI,

NDVI= (NIR-RED)/(NIR+RED)

This ratio yields the photosynthesis activity among each block of a particular region.

Task 4: Breaking of 1 pixel value(NDVI) value into 4 smaller region NDVI values

My findings: The bilinear interpolation code basically converts a image into a 2Dmatrix of different pixel value representing each colour. Instead of taking that input I took input of those pixel values given in input csv and applied the formula of bilinear interpolation and got new set of values.

Initial NDBI values:

0	0	0	0	0.102922	0.130542	0	0	0	0	0	0
0	0	0	0.065961	0.081583	0.075333	0.094388	0.129547	0.172261	0.175801	0	0
0	0	0	0.028844	0.022017	-0.02024	0.038317	0.143953	0.155822	0.168475	0.096462	0.137209
0	0	0	0.066243	-0.01634	0.006234	0.073952	0.161109	0.155822	0.132678	0.05135	0
0	0	0.0823	0.003181	-0.06075	-0.05966	0.036798	0.14379	0.083348	0.058476	0	0
0	0	0.069714	-0.00151	-0.02921	-0.04226	0.02228	0.059013	0.018522	0	0	0
0	-0.02062	0.045012	-0.00926	-0.00575	0.110379	0.069759	0.013557	0	0	0	0
0	-0.00019	0.0911	0.074531	0.082764	0.101662	0.016287	0	0	0	0	0
0	0.075517	0.162359	0.165339	0.107286	0.064663	0	0	0	0	0	0
0.113613	0.090776	0.136688	0.098507	0.022281	0.122347	0	0	0	0	0	0
0.087938	0.080785	0.089618	0.053659	0.075999	0	0	0	0	0	0	0
0	0	-0.02807	0	0	0	0	0	0	0	0	0

Output NDBI values:

0	0	0	0	0	0	0	0.021415	0.068584	0.106365	0.119024	0.125158	0.065331	0.005504	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.009613	0.024728	0.045313	0.072477	0.093585	0.098482	0.100639	0.075092	0.049544	0.054509	0.062566	0.071879	0.081667	0.086572	0.087383	0.073317	0.033033	0	0
0	0	0	0	0	0.019226	0.049456	0.069212	0.076371	0.080804	0.07794	0.076119	0.084852	0.093585	0.109019	0.125132	0.143759	0.163334	0.173144	0.174766	0.146635	0.066066	0	0
0	0	0	0	0	0.013817	0.035541	0.048318	0.050333	0.048777	0.037662	0.029148	0.046932	0.064717	0.095647	0.12791	0.14583	0.158338	0.16606	0.169771	0.151582	0.094796	0.049912	0.059249
0	0	0	0	0	0.008407	0.021627	0.027424	0.024295	0.016749	-0.002616	-0.017823	0.009012	0.035848	0.082276	0.130688	0.147902	0.153341	0.158976	0.164776	0.156528	0.123525	0.099824	0.118498
0	0	0	0	0	0.013858	0.035647	0.038242	0.017753	0.00161	-0.002899	-0.004398	0.024537	0.053473	0.096248	0.140426	0.153626	0.155134	0.154514	0.15211	0.137857	0.102719	0.073469	0.071039
0	0	0	0	0	0.019308	0.049667	0.049059	0.011211	-0.013529	-0.003182	0.009027	0.040062	0.071097	0.110221	0.150164	0.159349	0.156927	0.150052	0.139445	0.119185	0.081913	0.047114	0.02358
0	0	0	0.015427	0.034285	0.039273	0.036323	0.019469	-0.014105	-0.037072	-0.031648	-0.023327	0.014294	0.051914	0.095771	0.140259	0.141514	0.126453	0.113599	0.102597	0.08398	0.051944	0.023557	0.01179
0	0	0	0.030853	0.068571	0.059239	0.022979	-0.010121	-0.039421	-0.060616	-0.060115	-0.055681	-0.011475	0.032731	0.081321	0.130355	0.123679	0.095979	0.077147	0.065748	0.048775	0.021975	0	0
0	0	0	0.028494	0.063328	0.054096	0.019644	-0.008699	-0.029697	-0.045728	-0.048468	-0.047641	-0.010748	0.026145	0.059443	0.092378	0.08461	0.061482	0.045526	0.035582	0.024387	0.010988	0	0
0	0	0	0.026135	0.058085	0.048953	0.016309	-0.007277	-0.019972	-0.03084	-0.036821	-0.039601	-0.010021	0.019559	0.037566	0.054401	0.045541	0.026984	0.013904	0.005416	0	0	0	0
0	-0.004724	-0.009448	0.015061	0.046074	0.039072	0.010313	-0.007905	-0.013447	-0.011056	0.012564	0.034551	0.040033	0.045515	0.041969	0.037507	0.027294	0.014909	0.006952	0.002708	0	0	0	0
0	-0.009448	-0.018897	0.003987	0.034064	0.029192	0.004317	-0.008533	-0.006921	0.008728	0.061949	0.108704	0.090088	0.071472	0.046371	0.020614	0.009046	0.002833	0	0	0	0	0	0
0	-0.004768	-0.009537	0.019009	0.054967	0.057731	0.041497	0.033856	0.036548	0.046924	0.077865	0.103422	0.074551	0.045679	0.02794	0.01133	0.004523	0.001417	0	0	0	0	0	0
0	-8.82E-05	-0.000176	0.034032	0.075871	0.08627	0.078677	0.076244	0.080017	0.08512	0.093781	0.098141	0.059014	0.019886	0.009509	0.002045	0	0	0	0	0	0	0	0
0	0.017261	0.034521	0.071052	0.111871	0.124749	0.121635	0.114752	0.103336	0.093546	0.08811	0.080068	0.045687	0.011306	0.004755	0.001023	0	0	0	0	0	0	0	0
0	0.034609	0.069219	0.108073	0.147872	0.163227	0.164593	0.15326	0.126655	0.101972	0.082439	0.061996	0.032361	0.002726	0	0	0	0	0	0	0	0	0	0
0.056807	0.068878	0.08095	0.10803	0.13845	0.144393	0.136327	0.117954	0.087184	0.068364	0.081527	0.089648	0.046795	0.003942	0	0	0	0	0	0	0	0	0	0
0.113613	0.103147	0.092681	0.107988	0.129029	0.125559	0.108061	0.082647	0.047713	0.034755	0.080615	0.1173	0.061229	0.005159	0	0	0	0	0	0	0	0	0	0
0.100776	0.093903	0.087031	0.096042	0.108587	0.102348	0.085359	0.070477	0.058129	0.05064	0.056155	0.05865	0.030615	0.002579	0	0	0	0	0	0	0	0	0	0
0.087938	0.08466	0.081381	0.084096	0.088144	0.079136	0.062657	0.058307	0.068545	0.066525	0.031695	0	0	0	0	0	0	0	0	0	0	0	0	0
0.043969	0.04233	0.040691	0.036786	0.032377	0.029623	0.027816	0.029153	0.034273	0.033263	0.015848	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-0.010524	-0.02339	-0.01989	-0.007025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Task 5: Finding the no. of positive and negative values

My findings: According to NDBI if the value is less than 0 then there is vegetation and no build up found whereas the positive value of NDBI means there is a building present at that pixel.

I found the total positive and total negative pixel in the output csv and printed that for each satge and then we can analyse that there is a certain build up going on in a certain region.

2020 Data

```
before interpolation
Positive Count: 47
Negative Count: 22
after interpolation
Positive Count: 246
Negative Count: 104
```

2023 Data

```
before interpolation
Positive Count: 57
Negative Count: 12
after interpolation
Positive Count: 305
Negative Count: 45
```

Task 6: Plotting the new coordinates and create a cluster of the points

My findings:

I created a final file in which there are new latitude new longitudes and new pixel values.
Plot that on google earth.

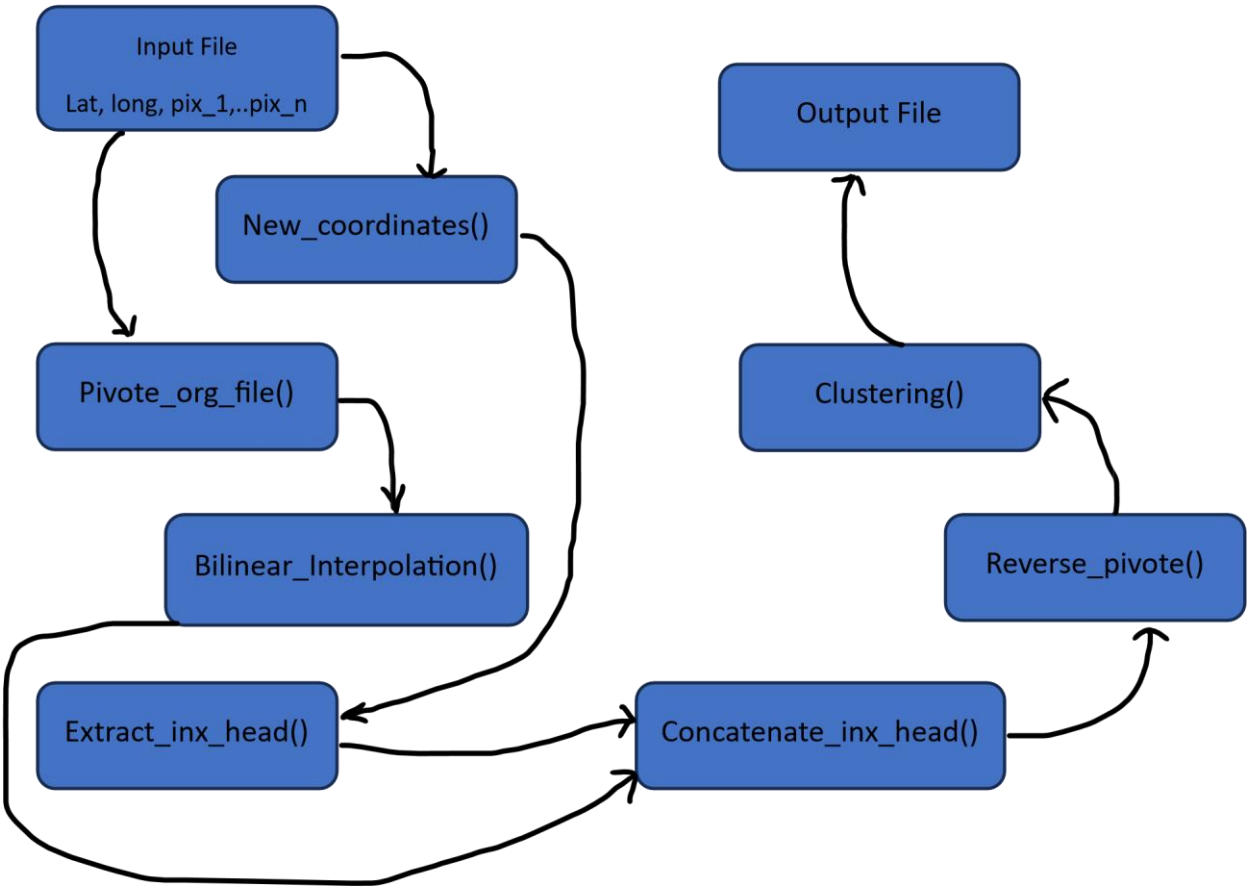
Clustering: I created 3 major clusters and ignoring the 0 values from the map

1.) -1 to 0.05 – The confirm build up
2.) 0.05 to 0.1 – Possible build up
3.) 0.1 to 1 – Vegetation

Output after clustering

Green dots- vegetation
Red dots- Confirm build up
Yellow dots- Possible build up

Task 7: Integrating all functions in 1 file (Main Code flow)



Task 7: Validation 1 – (Treeland)
Task 8: Validation 2 – (Kensington club) **wrong data**
Task 9: Validation 3 – (Sanika Building) **wrong data**

Task 10: Explore on NDBI

My findings: I found some irregularities in the input values of the NDBI value because on some date it was showing me complete vegetation and other day it was showing complete build up. That means there is some inconsistency in the input data. So I was searching for a generalize formula for the topic on the official documentation of sentinel satellite. There I found there was no such NDBI but a BU(Build- up index) was present which was giving more accurate results and a consistency was there in its values.

Formula of NDBI:

ndvi= (B05 - B04) / (B05 + B04);

ndbi = (B05 - B04) / (B05 + B04);

BU = (ndvi - ndbi)

Task 11: Data Extraction from Google Earth Engine.

Tools Used: Google Engine
Language: JavaScript
My Findings: As the data is not as accurate we need to test and compare the data from different satellites so I have to extract the data from my own rather tha depending on the third party.

I extracted the data from 2 satellites that are LANDSAT 9 and SENTINEL 2

	LANDSAT-9	SENTINEL-2 L2A
FORMULA	ndbi=(B6-B5)/(B6+B5)	ndbi=(B11-B8)/(B11+B8)
BANDS USED	B6 - SWIR 1 , B5 - NIR	B11 - SWIR 1 , B8 - NIR
PIXEL SIZE	30m x 30m	10m x10m
PIXEL VALUE	0.137093	0.034312,0.102835,-0.00518995,-0.0643275
PROBLEM IN IMPLEMENTATION	Higher pixel size need to interpolate the values and moderate accuracy	Very low accuracy and distorted images of the target location

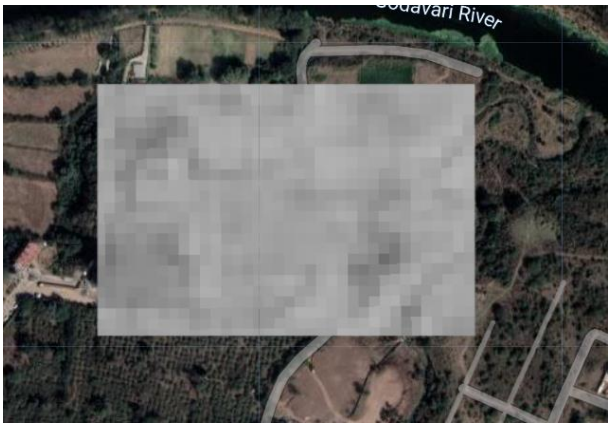
LANDSAT 9 IMAGE

SENTINEL 2 L2A IMAGE

ORIGNAL IMAGE

Task 12: Comparing NDVI and NDBI

NDBI	NDVI


Godavari River


Gangapur Rd


Gangapur Rd


Gangapur Rd