SC209 Project

Group 16

Under the able guidance of:

Prof. Ranendu Ghosh

Group Members:

1.	Margi Hingrajia	ID:201801014
2.	Shantanu Tyagi	ID:201801015
3.	Visaj Nirav Shah	ID:201801016
4.	Mitesh Koradia	ID:201801017
5.	Mahaveer Bohra	ID:201801018
6.	Rohan Jasoria	ID:201801019
7.	Kirtan Delwadia	ID:201801020
8.	Prayush Dave	ID:201801021
9.	Harshil Joshi	ID:201801022
10.	Arihant Pratap Singh	ID:201801023
11.	Akshay Mungalpara	ID:201801024
12.	Hemang Shrimali	ID:201801025
13.	Harshil Gandhi	ID:201801026
14.	Nishit Jagetia	ID:201801027
15.	Bhargav Patel	ID:201801465
16.	Siddhraj Parmar	ID:201801466
17.	Sudiksha Thusu	ID:201801469

Introduction

Over the last century, the average global surface temperatures have increased by 0.74 degrees. This warming has had a variety of impacts on all species and has affected their population, relative habitat, and their development. A major contributor to this climate change is carbon dioxide, as it is released by power plants, fires, vehicles, and many other processes. Through this project, we try to establish a relationship between climate change and phenology as well as atmospheric Carbon dioxide with phenology. Provided to us by Professor Ghosh, the objectives of this project were:

1. Understanding different satellite data and GIS tools.
2. Relating phenology with CO2 and climate change.
3. Using satellite data and ML model and predicting these variables and the possible impact on CO2 levels and climate.

We have used high-resolution satellite data for modeling and also consulted various research papers and articles. Also, we are happily able to deliver an ML model coded in Python along with this report.

Along the journey of this report, we would be exploring phenology, NDVI, EVI, their relationship with each other, the satellite and data retrieval method we used for this project followed by phenology's relation with atmospheric Carbon dioxide and then the final destination being our conclusions and interpretations.

We would like to thank professor Ghosh for giving us such an enlightening opportunity to work on a great project like this as a team. We would also like to thank the course teaching assistants for helping us and guiding us through every obstacle we came through during the making of this project.

PHENOLOGY

What is Phenology?

Phenology is the study of periodic plant and animal life cycle events and how these are influenced by seasonal and interannual variations in climate, as well as habitat factors.
It is the study of the timing of the biological events in plants and animals such as flowering, leafing, hibernation, reproduction, and migration.

Phenology is the study of the timing of recurring biological events, the interaction of biotic and abiotic forces that affect these events, and the interrelation among phases of the same or different species. Changes in weather with the seasons, such as temperature and precipitation, signal many organisms to enter new phases of their lives. Phenology is literally "the science of appearance." The word phenology comes from the Greek words phaino, meaning to show or appear, and logos, which means to study.

What does Phenology depend on?

Changes in the timing of phases of the plant life cycle, known as phenophases, are directly affected by temperature, rainfall, and day length. While these factors change through the year in places where there are distinct seasons, the first two – temperature and rainfall – are also changing in many regions because of climate change.
Leaf phenology depends primarily on the climatic conditions for a given biome. It strongly affects land-surface boundary conditions and the exchange of matter and energy with the atmosphere, influencing the surface albedo, roughness, and the dynamics of the terrestrial water cycle.

NDVI

What is NDVI?

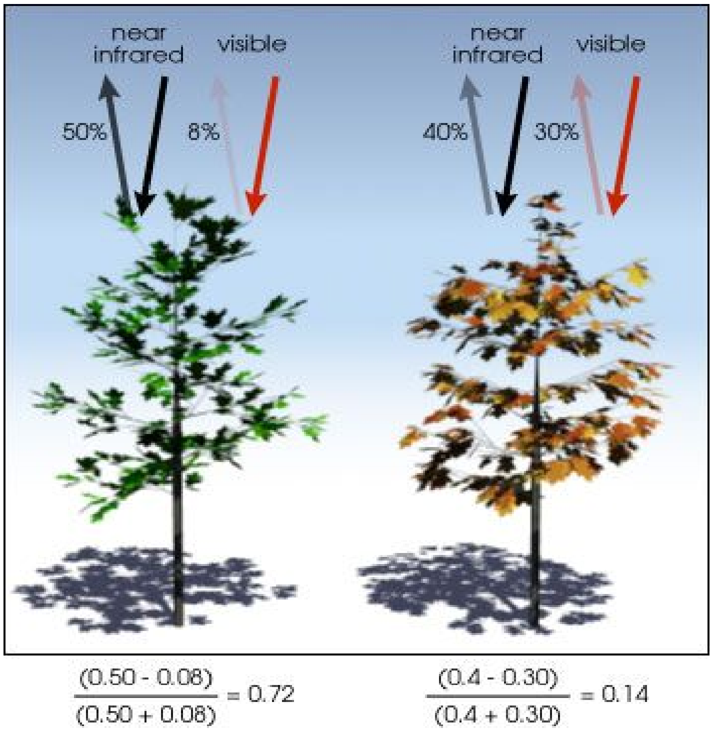
NDVI, expanded as Normalized Difference Vegetation Index, is a simple graphical indicator that can be used to analyze remote sensing measurements, often from a space platform, assessing whether or not the target being observed contains live green vegetation. NDVI is used around the world to monitor drought, forecast agricultural production, assist in forecasting fire zones and desert offensive maps.
Normalized Difference Vegetation Index (NDVI) quantifies vegetation by measuring the difference between near-infrared (which vegetation strongly reflects) and red light (which vegetation absorbs). NDVI ranges from -1 to +1 but there is no fixed boundary for the type of land cover.
For eg. when you have NDVI value very close to -1, it's most likely to be water while when you have NDVI value close to +1, then its high probability of dense green leaves. But when it's close to zero, there aren't any green leaves, and it could be a city.

How to Calculate NDVI?

NDVI is calculated from the visible and near-infrared light reflected by vegetation. Healthy vegetation absorbs most of the visible light that hits it and reflects a large portion of near-infrared light. Unhealthy or sparse vegetation reflects more visible light and less near-infrared light.
As we know, many different wavelengths make up the sunlight, when sunlight strikes an object, some wavelengths of the spectrum get reflected and some get absorbed. The pigment in plants leaves, named chlorophyll, strongly absorbs visible light (from 0.4 to 0.7 µm) for use in photosynthesis, while near-infrared lights (from 0.7 to 1.1 µm) get reflected. So NDVI can be calculated by taking the ratio of difference of absorbed and reflected light to the total light reaching the surface.

$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$

Healthy vegetation (chlorophyll) reflects more near-infrared (NIR) and green light compared to other wavelengths. But it absorbs more red and blue light.



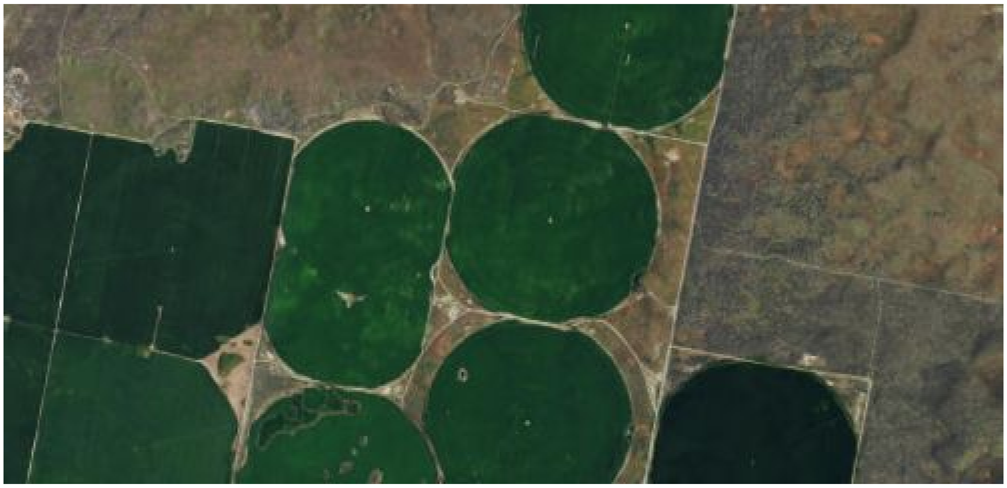
$$\frac{(0.50 - 0.08)}{(0.50 + 0.08)} = 0.72 \qquad \frac{(0.4 - 0.30)}{(0.4 + 0.30)} = 0.14$$

This is why our eyes see vegetation as the **color green**. If you could see near-infrared, then it would be strong for vegetation too. Satellite sensors like Landsat and Sentinel-2 both have the necessary bands with NIR and red.

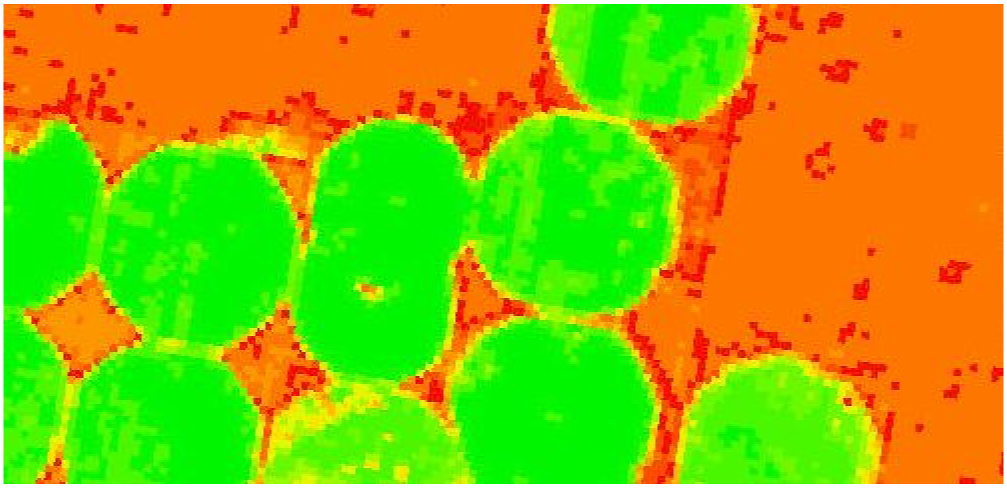
The result of this formula generates a value of -1 to +1. Higher the NDVI value means healthier the vegetation of that area and vice versa.

Let’s see an example of NDVI over agricultural land -

When an image is taken of agricultural land for red, green, blue band, it looks as our eyes see.



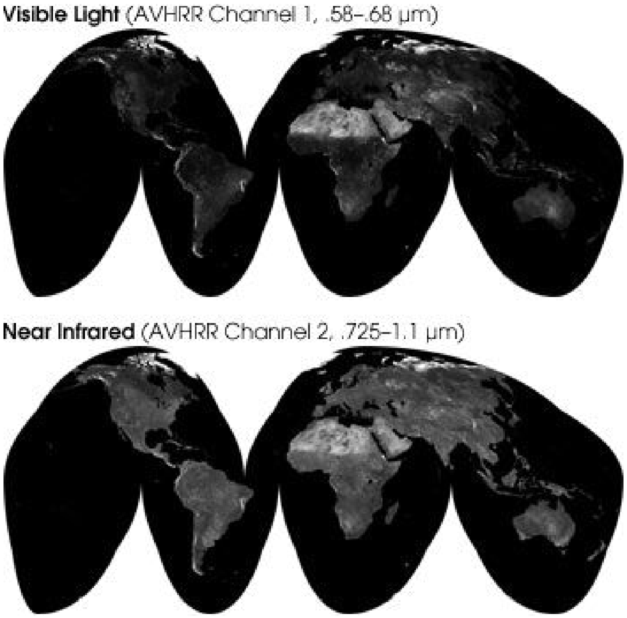
When we apply the formula, bright green indicates high NDVI whereas red indicates low NDVI. So we can quantify vegetation by taking the difference of near-infrared (which vegetation reflects) and red (which vegetation absorbs).



Satellites and Devices for NDVI

Satellites like Sentinel-2, Landsat and SPOT produce red and near-infrared images. The NOAA AVHRR instrument has five detectors, two of which are sensitive to the wavelengths of light ranging from 0.55–0.70 and 0.73–1.0 micrometers.

With these detectors, one can measure the intensity of light coming off the earth in visible and near-infrared wavelengths. From these intensities, we can quantify the photosynthetic capacity of vegetation in a given pixel of the land surface. In general, if there is much more reflected radiation in near-infrared than in visible wavelength, then vegetation in that pixel is quite good. So in this way, we can calculate vegetation capacity for all pixels of an image to know the vegetation index of a land area.



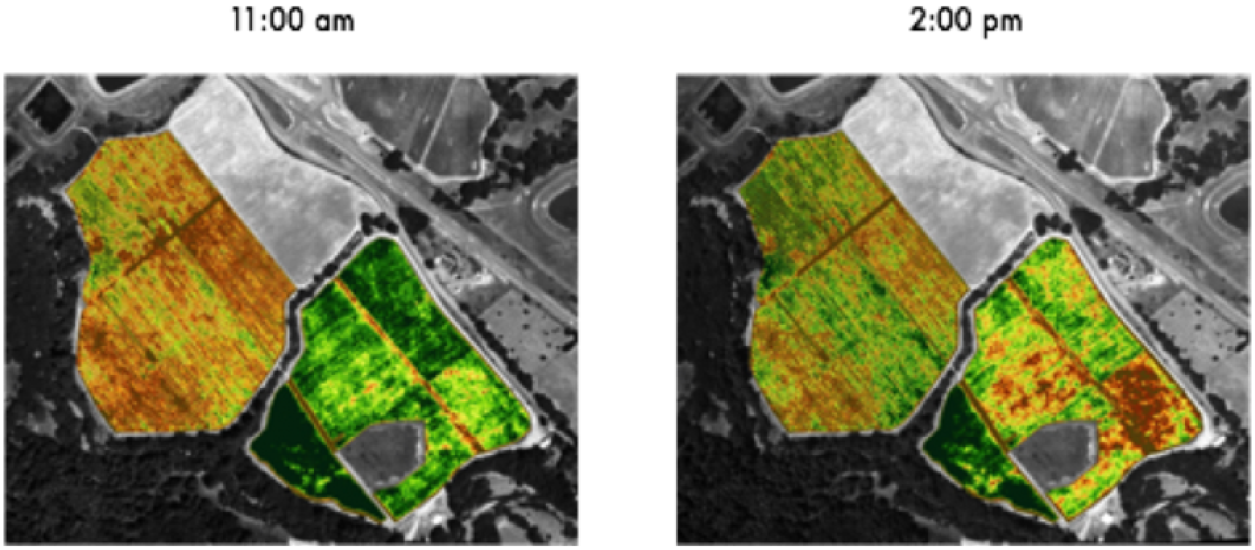
Visible Light (AVHRR Channel 1, .58–.68 µm)

Near Infrared (AVHRR Channel 2, .725–1.1 µm)

Shortcomings of NDVI

NDVI has been the standard to measure the vegetation index of a land area. What many do not know is that NDVI was not designed to measure plant vigor. It was designed to simply detect living vegetation and distinguish it from other matter like rocks, soil, or dead vegetation.**NDVI is loaded with potential errors and inaccuracies.**

- One of the most important shortcomings of NDVI is its dependency on the time of day at which it takes images. As NDVI does not correct for changes in the angle at which sun shines on the leaves, NDVI produces very different results throughout the day even though the health of the vines has obviously not changed in just a couple of hours.



- NDVI also produces inaccurate images due to a lot of factors like clouds, air moisture, shadow and variation in soil.

In order to overcome these inaccuracies, researchers come up with an advanced version of the vegetation index which uses additional wavelengths of lights in order to eliminate these errors of NDVI. This was the **Enhanced Vegetation Index (EVI).**

EVI

What is EVI?

Enhanced Vegetation Index(EVI), as the name suggests, is an enhanced version of vegetation index(VI). As we know, a Vegetation Index (**VI**) is a spectral transformation of two or more bands designed to enhance the contribution of vegetation properties and allow reliable spatial and temporal inter-comparisons of terrestrial photosynthetic activity and canopy structural variations.
It was developed as an alternative vegetation index in order to address some of the limitations of NDVI. The EVI was specifically developed to :

1. Be more sensitive to changes in areas having high biomass
2. Reduce the influence of atmospheric conditions on vegetation index values, and

3. Correct for canopy background signals.

EVI is more sensitive to plant canopy differences like leaf area index(LAI), canopy structure and plant phenology and stress than NDVI which generally respond to the amount of chlorophyll present in that area. Briefly, Moderate Resolution Imaging Spectroradiometer (MODIS) is a sensor that provides much higher spatial resolution.

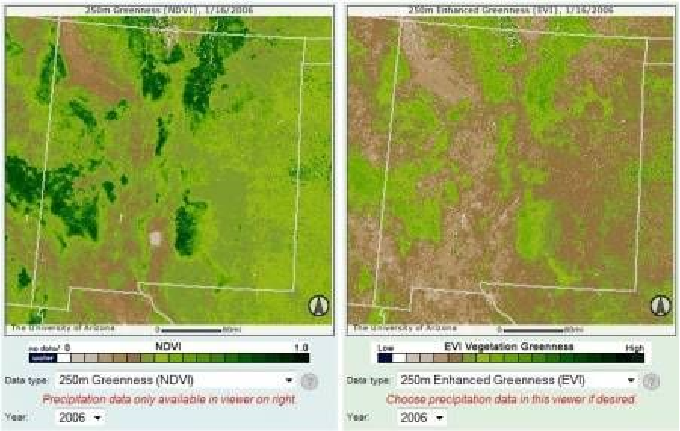
How to calculate EVI?

EVI is the same as NDVI but just a better version with fewer errors. So it is calculated with the same logic as of NDVI but to remove some errors, some additional wavelengths are added to it. EVI is calculated as

$$EVI = 2.5 * \frac{(NIR - RED)}{(NIR + C_1 * RED - C_2 * BLUE + L)}$$

where NIR, RED, and BLUE are atmospherically-corrected (or partially atmospherically-corrected) surface reflectances, and C_1, C_2, and L are coefficients to correct for atmospheric condition (i.e., aerosol resistance). For the standard MODIS EVI product, L=1, C_1=6, and C_2=7.5.

The EVI takes full advantage of MODIS' new, state-of-the-art measurement capabilities. While the EVI is calculated similarly to NDVI, it corrects for some distortions in the reflected light caused by the particles in the air as well as the ground cover below the vegetation. The EVI data product also does not become saturated as easily as the NDVI when viewing rainforests and other areas of the Earth with large amounts of chlorophyll.


250m Greenness (NDVI), 1/16/2006
250m Enhanced Greenness (EVI), 1/16/2006
The University of Arizona
0
80mi
no data/ cooler
0
NDVI
1.0
Low
EVI Vegetation Greenness
High

Data type: 250m Greenness (NDVI)
Precipitation data only available in viewer on right
Year: 2006

Data type: 250m Enhanced Greenness (EVI)
Choose precipitation data in this viewer if desired
Year: 2006

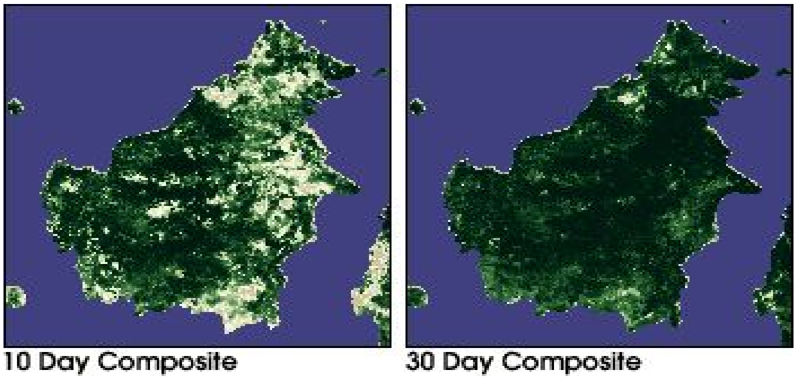
MODIS EVI (right) compared to NDVI (left) for a region over the same period. The NDVI image shows a greater area in dark green because NDVI loses sensitivity to changes in vegetation in areas of rainforest, higher biomass. The EVI image maintains a more consistent sensitivity to change in vegetation and it has a more even distribution of vegetation greenness values.

Satellites and Devices for EVI

With the launch of the two MODIS sensors on Terra (satellite) and Aqua in 1999, NASA adopted EVI as a standard MODIS product that is distributed by the USGS. MODIS will provide images over a given pixel of land just as often as AVHRR but in much finer detail and with measurements in a greater number of wavelengths using detectors that were specifically designed for measurements of land surface dynamics.

MODIS Vegetation Index Product

Neither NDVI nor EVI products can eliminate all obstacles. Clouds and aerosols can often block the satellites' view of the surface entirely, direct rays from the sun can sometimes saturate certain pixels and sometimes disturbance in the satellite itself can distort an image. So long term averages of vegetation data are taken to help remove as much error as possible. The below pair of images shows the difference between long term avg. and short term avg.



10 Day Composite

30 Day Composite

MODIS vegetation indices, produced on 16-day intervals and at multiple spatial resolutions, provide consistent spatial and temporal comparisons of vegetation canopy greenness, a composite property of leaf area, chlorophyll and canopy structure. Two vegetation indices are derived from atmospherically corrected wavelengths in red, near-infrared (NDVI) and blue wavelength (EVI). The Normalized Difference Vegetation Index (NDVI), which provides continuous data using NOAA's AVHRR record for historical and climate applications and the Enhanced Vegetation Index (EVI), which minimize canopy soil variation and improves sensitivity over dense vegetation regions. The two products more effectively characterize the global range of vegetation states.

The vegetation indices are retrieved daily from atmosphere corrected, bidirectional reflection from the surface. MODIS uses a method based on quality assurance metrics to remove low-quality pixels. From the remaining good quality VI values, a constrained view angle approach then selects a pixel to represent the compositing period (from the two highest NDVI values it selects the pixel that is closest-to-nadir). Because the MODIS sensors aboard Terra and Aqua satellites are identical, the VI algorithm generates each 16-day composite eight days apart (phased products) to permit a higher temporal resolution product by combining both data records. The MODIS VI product suite is now used successfully in all ecosystems, climate, and natural resources management studies and operational research as demonstrated by the ever-increasing body of peer publications.

Data Collection (NDVI and EVI)

Google Earth Engine(GEE)

Google earth engine is a planetary-scale platform for Earth science data & analysis. It is powered by Google's cloud infrastructure. Google earth engine is used for collecting **High resolution of satellite data for modeling.** It is the most advanced cloud-based geospatial processing platform in the world
The easily accessible and user-friendly front-end provides a convenient environment for interactive data and algorithm development

Google Earth Engine is used in this project to collect the satellite data from different regions of **Gujarat** at different durations. GEE has large dataset archives which were used for NDVI and EVI data collection. The code in the GEE code editor is written in javascript.

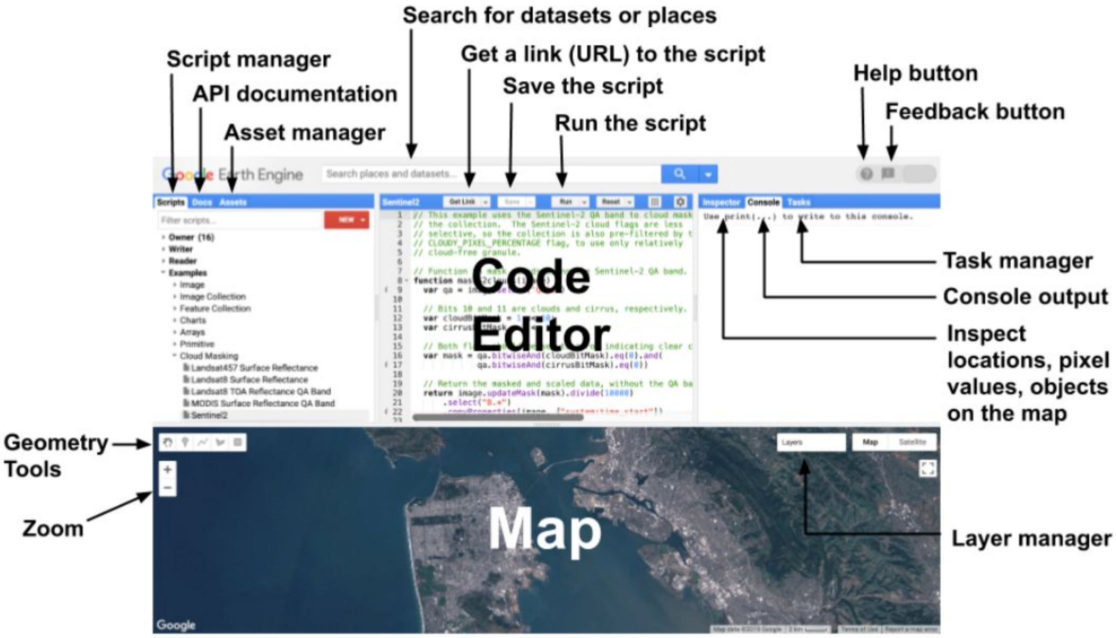
Components

The main components of Google Earth Engine are-

- **Datasets:** Google Earth Engine has Ready-to-Use Datasets. The public data archive includes more than thirty years of historical imagery and scientific datasets that are updated daily. It contains over twenty petabytes of geospatial data available for analysis.

- **APIs:** The Earth Engine API is available in Python and JavaScript, making it easy to harness the power of Google's cloud for your own geospatial analysis.

- **Code Editor:** The code editor is an online Integrated Development Environment (IDE) for rapid prototyping and visualization of complex spatial analyses using the Javascript API.



```
// This example uses the Sentinel-2 QA band to cloud mask
// the collection.  The Sentinel-2 cloud flags are less
// selective, so the collection is also pre-filtered by t
// CLOUDY_PIXEL_PERCENTAGE flag, to use only relatively
// cloud-free granule.

// Function [illegible] Sentinel-2 QA band.
function mas[illegible]
  var qa = im[illegible]

  // Bits 10 and 11 are clouds and cirrus, respectively.
  var cloudB[illegible]
  var cirrus[illegible]

  // Both fl[illegible] indicating clear c
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0).and(
             qa.bitwiseAnd(cirrusBitMask).eq(0))

  // Return the masked and scaled data, without the QA ba
  return image.updateMask(mask).divide(10000)
      .select("B.*")
```

```
Use print(...) to write to this console.
```

Datasets used in this Study-

MODIS(MODERATE RESOLUTION IMAGING SPECTRORADIOMETER)

The Moderate Resolution Imaging Spectroradiometer (MODIS) sensors on NASA's Terra and Aqua satellites have been acquiring images of the Earth daily since 1999, including daily imagery, 16-day BRDF-adjusted surface reflectance, and derived products such as vegetation indices and snow cover

1) MYD13A1.006 Aqua Vegetation Indices 16-Day Global 500m (MODIS)

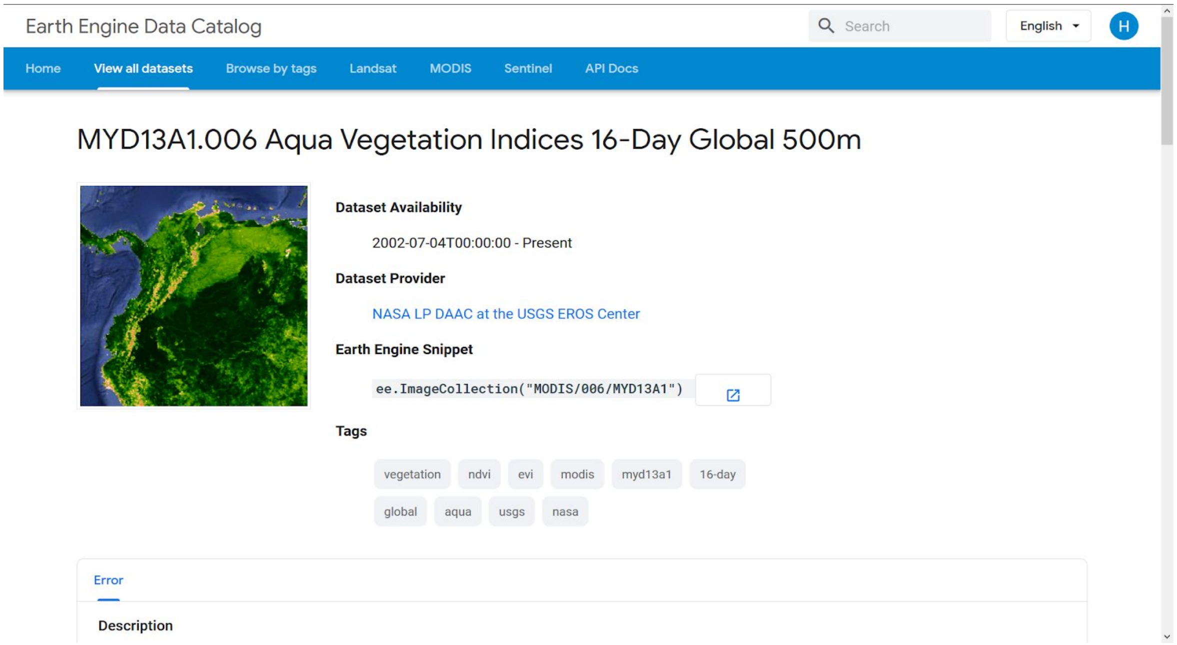
This Dataset is used for **NDVI and EVI** data collection is

MYD13A1.006 Aqua Vegetation Indices 16-Day Global 500m (MODIS)
Dataset Availability - 2002-07-04T00:00:00 - Present
Dataset Provider - NASA LP DAAC at the USGS EROS CENTER
Resolution - 500 meters
Time Frame - 1 January 2017- 30 August 2019
(Used for this project)

The MYD13A1 V6 product provides a Vegetation Index (VI) value on a per-pixel basis. There are two primary vegetation layers.

The first is the **Normalized Difference Vegetation Index (NDVI)** which is referred to as the continuity index to the existing National Oceanic and Atmospheric Administration-Advanced Very High-Resolution Radiometer (NOAA-AVHRR) derived NDVI.

The second vegetation layer is the **Enhanced Vegetation Index (EVI)** that minimizes canopy background variations and maintains sensitivity over dense vegetation conditions. The EVI also uses the blue band to remove residual atmosphere contamination caused by smoke and sub-pixel thin cloud clouds. The MODIS NDVI and EVI products are computed from atmospherically corrected bi-directional surface reflectances that have been masked for water, clouds, heavy aerosols, and cloud shadows.



MYD13A1.006 Aqua Vegetation Indices 16-Day Global 500m

Dataset Availability

2002-07-04T00:00:00 - Present

Dataset Provider

NASA LP DAAC at the USGS EROS Center

Earth Engine Snippet

```
ee.ImageCollection("MODIS/006/MYD13A1")
```

Tags

vegetation ndvi evi modis myd13a1 16-day global aqua usgs nasa

Error

Description

2) MOD11A2.006 Terra Land Surface Temperature and Emissivity 8-Day Global 1km

The MOD11A2 V6 product provides an average 8-day land surface temperature (LST) in a 1200 x 1200 kilometer grid. Each pixel value in MOD11A2 is a simple average of all the corresponding MOD11A1 LST pixels collected within that 8 day period. The 8 day compositing period was chosen because twice that period is the exact ground track repeat period of the Terra and Aqua platforms.

This Dataset is used for the collection of **Temperature Data.**

MOD11A2.006 Terra Land Surface Temperature and Emissivity 8-Day Global 1km

Dataset Availability	**-**	2000-03-05T00:00:00 - Present
Dataset Provider	**-**	NASA LP DAAC at the USGS EROS CENTER
Resolution	-	1000 meters
Time Frame (Used for this project)	**-**	1 January 2019- 1 January 2020

Dataset link

3) TRMM 3B43: Monthly Precipitation Estimates

This dataset algorithmically merges microwave data from multiple satellites, including SSMI, SSMIS, MHS, AMSU-B and AMSR-E, each inter-calibrated to the TRMM Combined Instrument.

Algorithm 3B43 is executed once per calendar month to produce the single, best-estimate precipitation rate and RMS precipitation-error estimate field (3B43) by combining the 3-hourly merged high-quality/IR estimates (3B42) with the monthly accumulated Global Precipitation Climatology Centre (GPCC) rain gauge analysis.

This Dataset is used for the collection of **Precipitation Data.**

TRMM 3B43: Monthly Precipitation Estimates

Dataset Availability	**-**	1998-01-01T00:00:00 - Present
Dataset Provider	**-**	NASA GSFC
Resolution	-	0.25 arc degrees
Time Frame (Used for this project)	**-**	1 January 2019- 1 January 2020

Dataset link

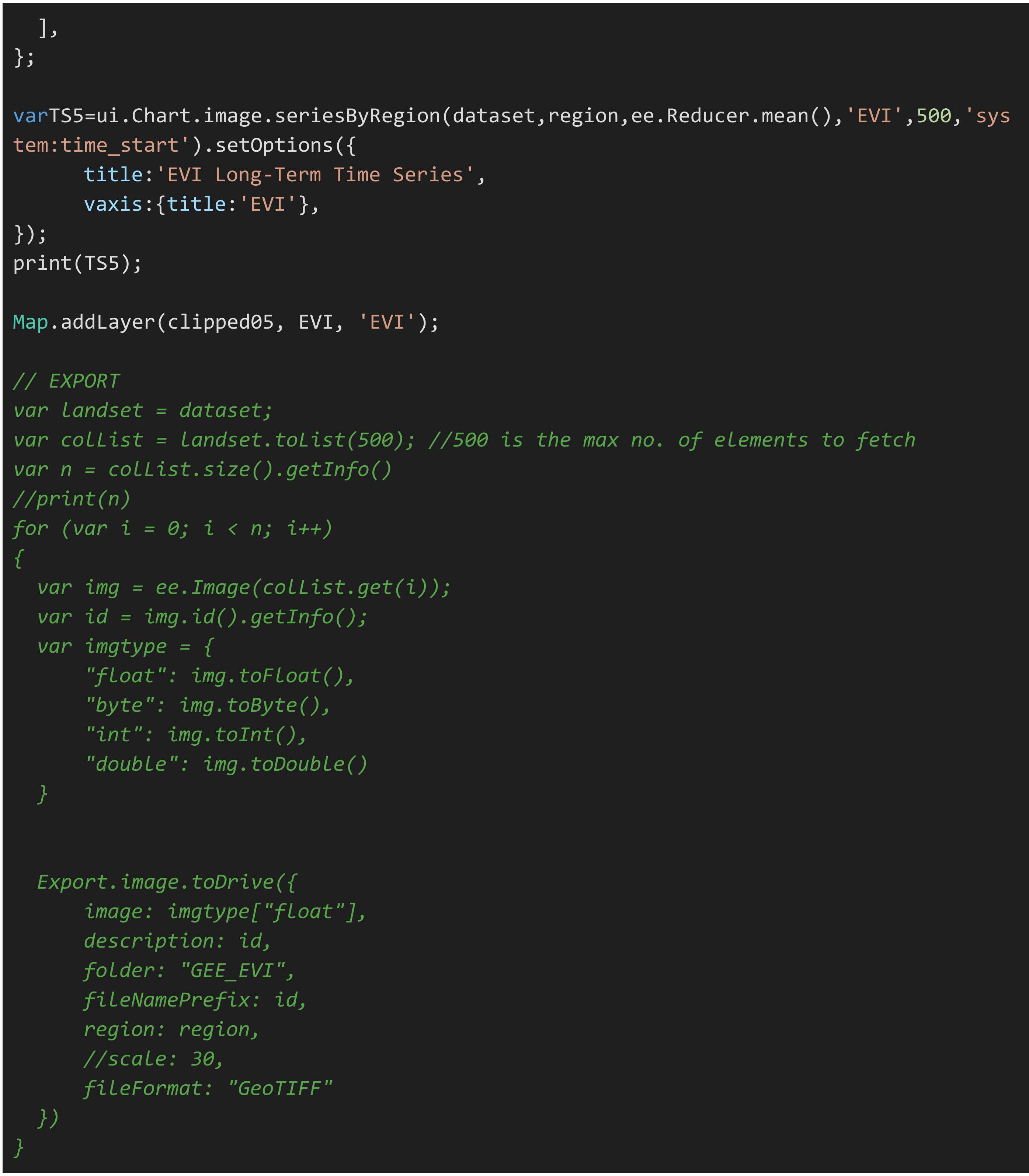
CODES used for Data Collection

NDVI (code link)

```
// NDVI Data collection for coordinate (23,70.5)
var region = /* color: #d63000 */ee.Geometry.Point([70.5, 23]);
var dataset = ee.ImageCollection('MODIS/006/MYD13A1')
                  .filter(ee.Filter.date('2010-01-01', '2019-08-30'))
                  .sort('CLOUD_COVER',true)
                  .select('NDVI');

var clipped05=dataset.mean().clip(region);

var ndviVis = {
  min: 0.0,
  max: 9000.0,
  palette: [
      'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
      '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
      '012E01', '011D01', '011301'
  ],
};

var
TS5=ui.Chart.image.seriesByRegion(dataset,region,ee.Reducer.mean(),'NDVI',500,'syst
em:time_start').setOptions({
      title:'NDVI Long-Term Time Series',
      vaxis:{title:'NDVI'},
});
print(TS5);

Map.addLayer(clipped05, ndviVis, 'NDVI');

// EXPORT
var landset = dataset;
var colList = landset.toList(500); //500 is the max no. of elements to fetch
var n = colList.size().getInfo()
//print(n)
for (var i = 0; i < n; i++)
{
  var img = ee.Image(colList.get(i));
  var id = img.id().getInfo();
  var imgtype = {
      "float": img.toFloat(),
      "byte": img.toByte(),
      "int": img.toInt(),
      "double": img.toDouble()
  }


  Export.image.toDrive({
      image: imgtype["float"],
      description: id,
      folder: "GEE_NDVI",
      fileNamePrefix: id,
      region: region,
      //scale: 30,
      fileFormat: "GeoTIFF"
  })
}
```

EVI (code link)

```
// EVI Data collection for coordinate (23,70.5)
var region = /* color: #d63000 */ee.Geometry.Point([70.5, 23]);
var dataset = ee.ImageCollection('MODIS/006/MYD13A1')
                  .filter(ee.Filter.date('2010-01-01', '2019-08-30'))
                  .sort('CLOUD_COVER',true)
                  .select('EVI');

var clipped05=dataset.mean().clip(region);

var EVI = {
  min: 0.0,
  max: 9000.0,
  palette: [
      'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
      '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
      '012E01', '011D01', '011301'
```



```
  ],
};

varTS5=ui.Chart.image.seriesByRegion(dataset,region,ee.Reducer.mean(),'EVI',500,'sys
tem:time_start').setOptions({
      title:'EVI Long-Term Time Series',
      vaxis:{title:'EVI'},
});
print(TS5);

Map.addLayer(clipped05, EVI, 'EVI');

// EXPORT
var landset = dataset;
var colList = landset.toList(500); //500 is the max no. of elements to fetch
var n = colList.size().getInfo()
//print(n)
for (var i = 0; i < n; i++)
{
  var img = ee.Image(colList.get(i));
  var id = img.id().getInfo();
  var imgtype = {
      "float": img.toFloat(),
      "byte": img.toByte(),
      "int": img.toInt(),
      "double": img.toDouble()
  }


  Export.image.toDrive({
      image: imgtype["float"],
      description: id,
      folder: "GEE_EVI",
      fileNamePrefix: id,
      region: region,
      //scale: 30,
      fileFormat: "GeoTIFF"
  })
}
```

Collection of CO_2 data:

We collected CO_2 data from NOAA's (National Oceanic and Atmospheric Administration) satellite data.

NOAA is an American scientific agency within the United States Department of Commerce that focuses on the conditions of the oceans, major waterways, and the atmosphere.

NOAA warns of dangerous weather, charts seas, guides the use and protection of ocean and coastal resources and conducts research to provide the understanding and improve stewardship of the environment.

CarbonTracker is a CO_2 measurement and modeling system developed by NOAA to keep track of sources and sinks of CO_2 around the world. CarbonTracker uses atmospheric CO2 observations from a host of collaborators and simulated atmospheric transport to estimate these surface fluxes of CO2.

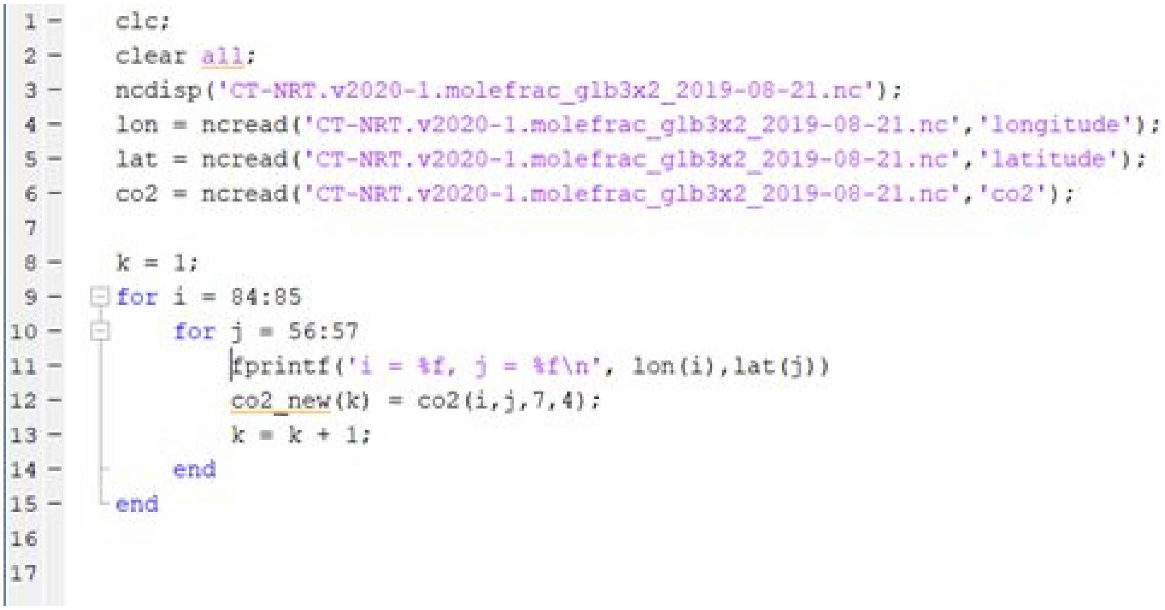
NOAA has stored data of the level of CO_2 (in µmol per mol) at 10800 coordinates of Earth on each day in different NetCDF4 files. In each file, they've stored 3 hourly data of 10800 coordinates on Earth on a given day.

Since our area of interest was Gujarat, we found that only 4 of these 10800(120(Longitude) * 90(Latitude)) coordinates were of any relevance to us. For that, we wrote a Matlab code to extract those 4 point's data. The 4 points were:

- 21°N 70.5°E
- 21°N 73.5°E
- 23°N 70.5°E
- 23°N 73.5°E

Our training set consisted of 111 day's data, and our test set consisted of 111 day's data. Data corresponding to these days was collected, and then we stored that data in a csv (Comma-separated values) file.

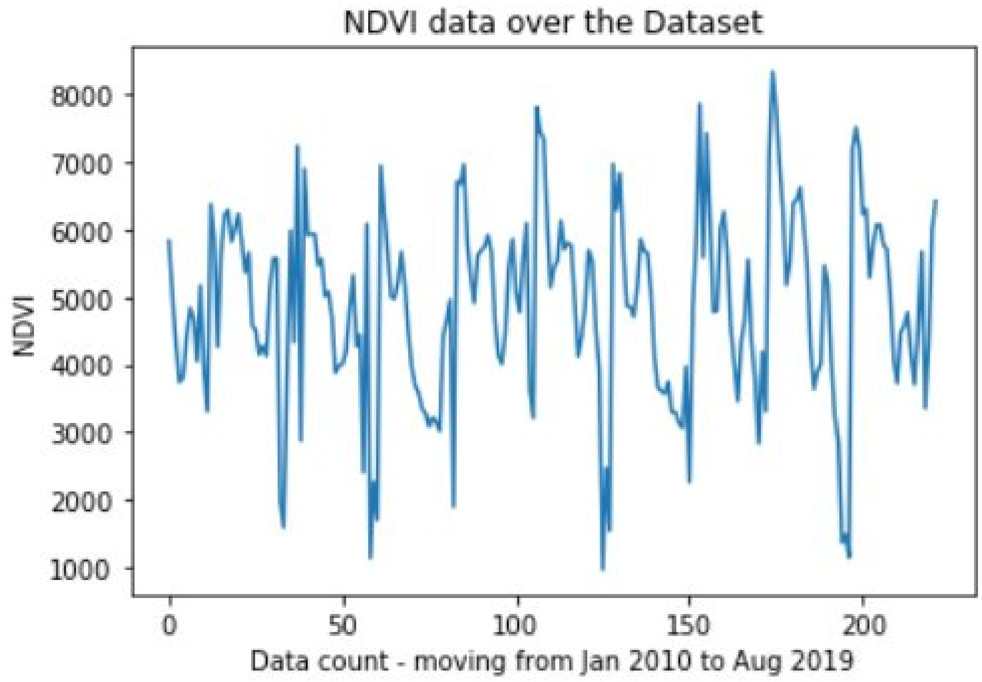
The Matlab code to read through each NetCDF4 file is shown in the picture below:

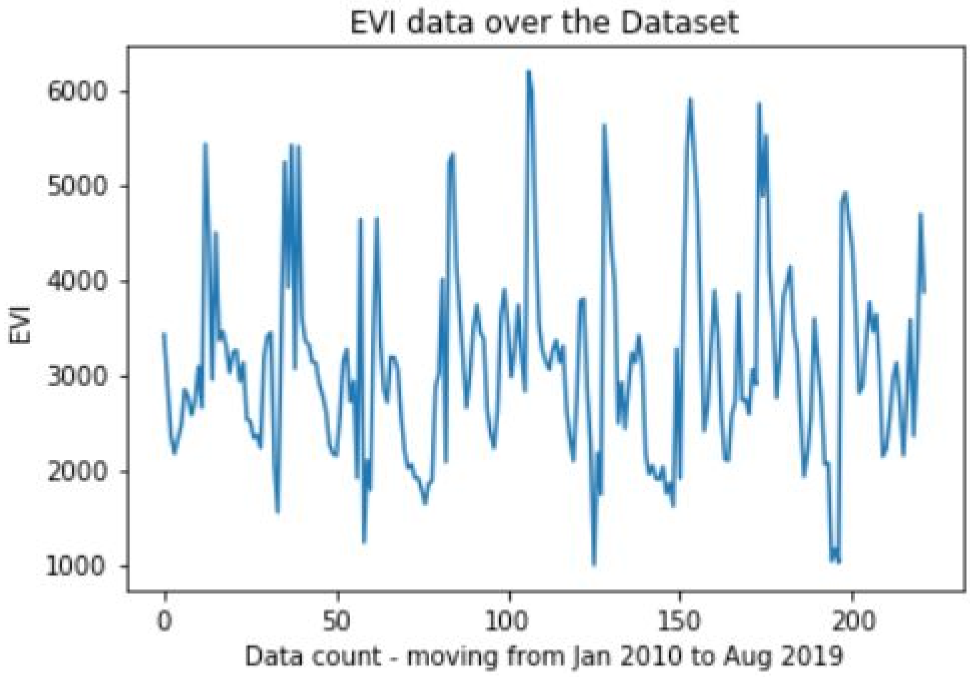


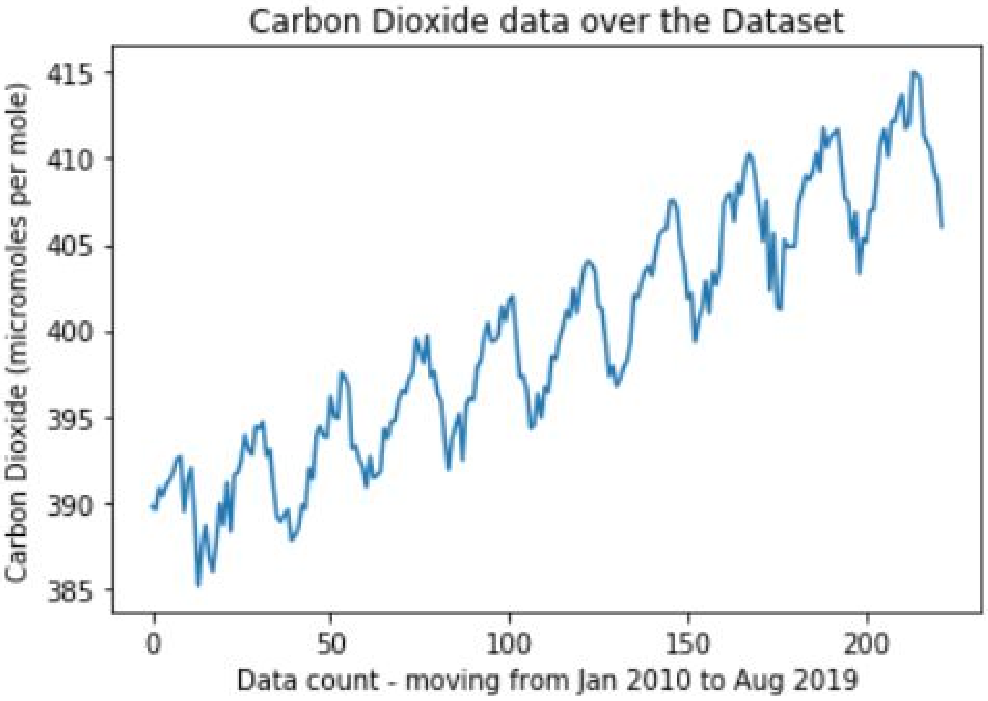
```
clc;
clear all;
ncdisp('CT-NRT.v2020-1.molefrac_glb3x2_2019-08-21.nc');
lon = ncread('CT-NRT.v2020-1.molefrac_glb3x2_2019-08-21.nc','longitude');
lat = ncread('CT-NRT.v2020-1.molefrac_glb3x2_2019-08-21.nc','latitude');
co2 = ncread('CT-NRT.v2020-1.molefrac_glb3x2_2019-08-21.nc','co2');

k = 1;
for i = 84:85
    for j = 56:57
        fprintf('i = %f, j = %f\n', lon(i),lat(j))
        co2_new(k) = co2(i,j,7,4);
        k = k + 1;
    end
end
```

Understanding Data

Trends of CO_2 and NDVI and EVI in yearly data:







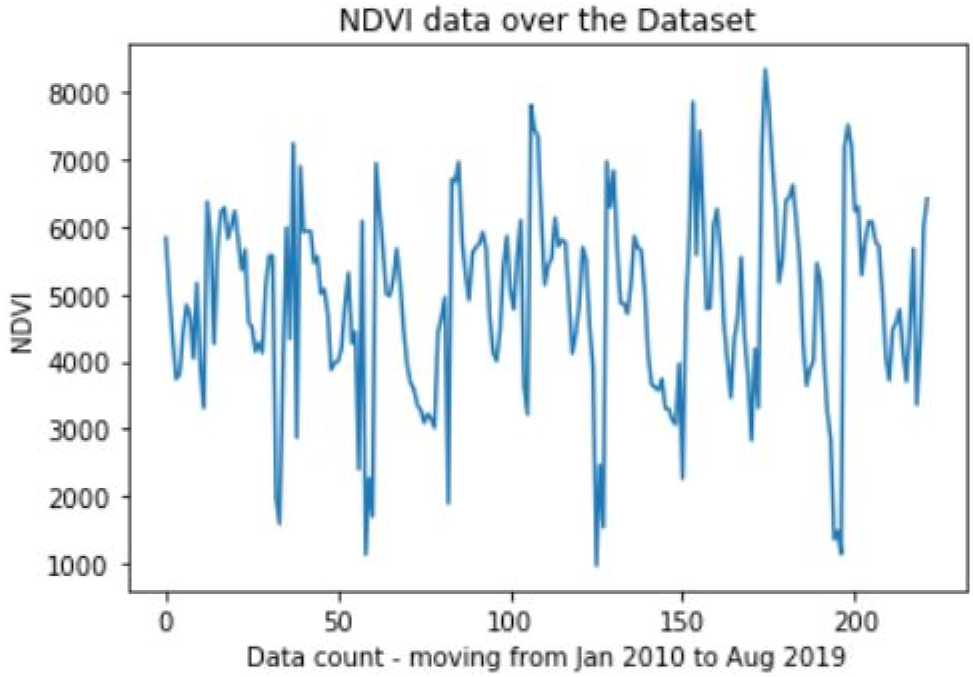
OBSERVATION:

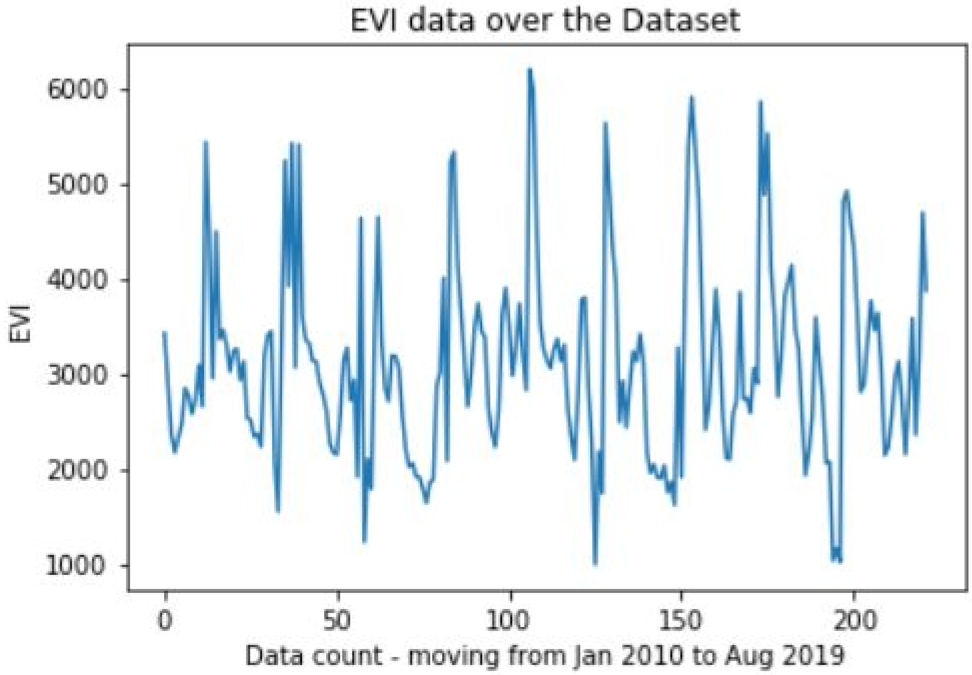
From the above-mentioned graphs, we can observe that the value of CO_2 increases as the value of NDVI and EVI decreases.

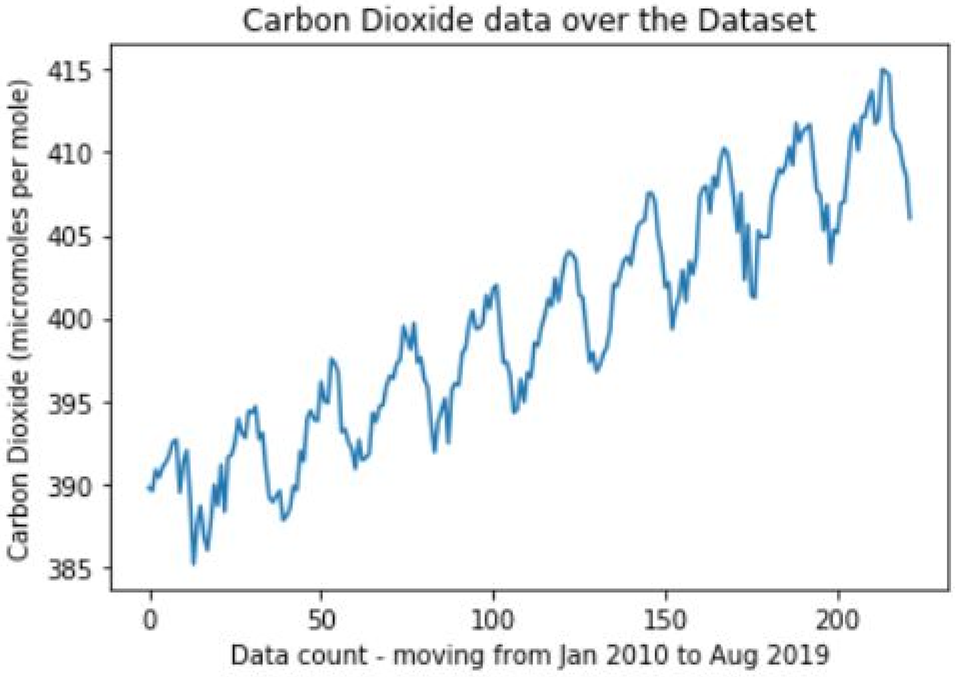
CONCLUSION:

As the value of NDVI or EVI increases, we can tell that the state of plant health in that region is quite good. If the state of plant health is good then we can safely say that the chlorophyll levels are quite high, which means that the process of Photosynthesis can be carried out in a more effective way. Due to more effective Photosynthesis, the CO_2 is consumed, which can be seen in the bottom-most graph above.

Trends of CO_2 and NDVI and EVI in the Whole Data-set:







OBSERVATION:

We observe that the extremum values(maximum and minimum) of NDVIand EVI don’t change in a year, whereas the values of CO_2 have a step function-esque shape.

CONCLUSION:

From this, we can say that CO_2 can't be solely predicted by NDVI and EVI, because they don't produce the change which makes CO_2 climb. But we can surely say that the NDVI and EVI data of one year and corresponding levels of CO_2 may have a relation from our last conclusion (for the topic 'Trends of CO_2 and NDVI and EVI in Whole Data-set'). Therefore, we can conclude that NDVI and EVI aren't the only factors affecting the CO_2 levels, and we can also conclude at the same time that NDVI and EVI values do affect CO_2 levels.

Predicting CO2 with NDVI and EVI:

Now that we have collected the data and analyzed it, it is time to find a relationship between these variables.

We will express Carbon Dioxide as a function of NDVI and EVI (phenological factors).

To find this function, we use Machine Learning algorithms.

A Brief Overview of Machine Learning:-

According to Wikipedia, Machine learning is the study of computer algorithms that improve automatically through experience. We create such algorithms to help the computer learn by analyzing pre-recorded data.

Here, for our research question, we will use a Machine Learning technique known as Polynomial Regression.

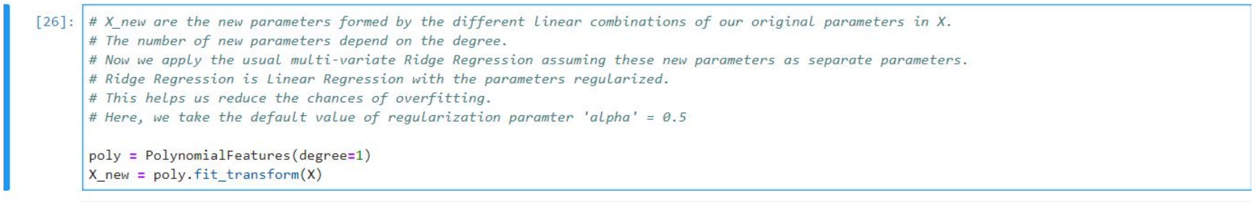
Before we begin, let's define the independent and dependent variables. Dependent variables are what we predict using independent variables. Here, NDVI and EVI (of 4 points whose data we have) are the independent variables and Carbon Dioxide is the dependent variable.

Hence, we have 8 independent variables. For this reason, since we have more than 1 independent variable, we will use the Multivariate Polynomial Regression of different degrees.

What is Polynomial Regression?

Polynomial Regression is a machine learning method in which the dependent variable is represented as a polynomial function of independent variables.
The degree of Polynomial Regression is the highest degree of the individual terms of the polynomial. For example, (A + B*X) has degree 1, (A + B*X + C*X^2) has degree 2, and so on. (X is the variable.)
To train our model, we will use the data available from Jan 2010 to August 2019. The frequency of data is 32 days.

We will start with **Polynomial Regression with Degree 1**. This is also known as Linear Regression.



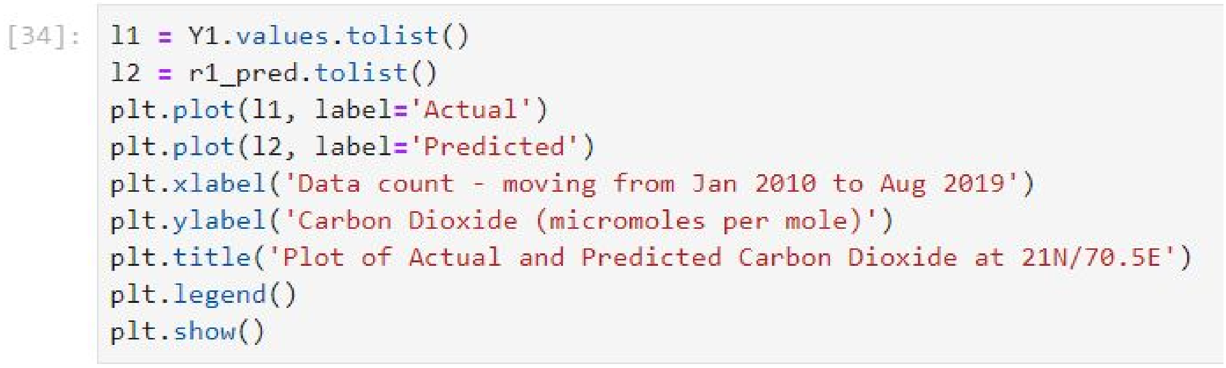
```
[26]: # X_new are the new parameters formed by the different linear combinations of our original parameters in X.
      # The number of new parameters depend on the degree.
      # Now we apply the usual multi-variate Ridge Regression assuming these new parameters as separate parameters.
      # Ridge Regression is Linear Regression with the parameters regularized.
      # This helps us reduce the chances of overfitting.
      # Here, we take the default value of regularization paramter 'alpha' = 0.5

      poly = PolynomialFeatures(degree=1)
      X_new = poly.fit_transform(X)
```

Once we have created a PolynomialFeatures object and transformed our original parameters, we create RidgeRegression objects (1 for each coordinate point). 0

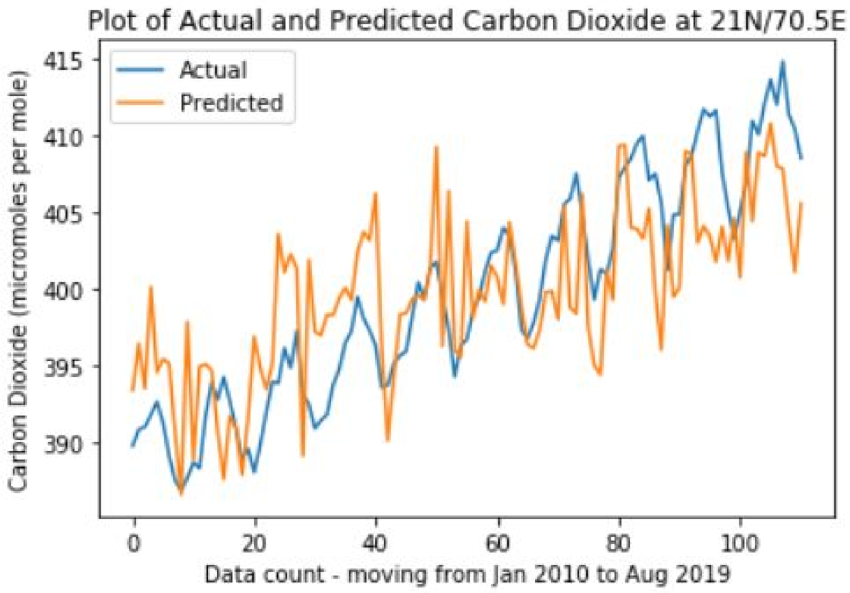
We then train the model (fit the data). Using this, we predict the values of Carbon Dioxide (dependent variable).

Once we have the predicted values, we evaluate the performance of our model. We do this in two ways, through graphs (qualitative) and R-squared score and MSE (quantitative).

First, let us look at the graphs.

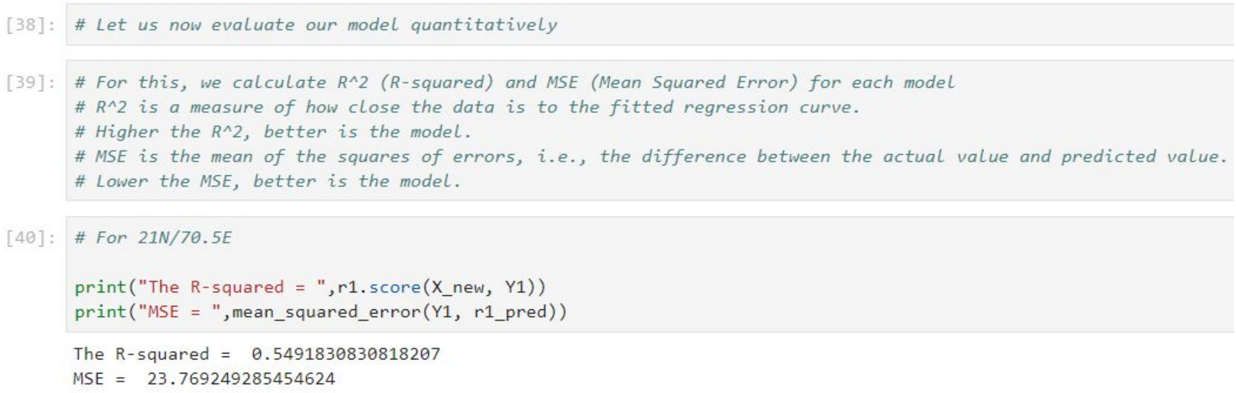


```
[34]: l1 = Y1.values.tolist()
      l2 = r1_pred.tolist()
      plt.plot(l1, label='Actual')
      plt.plot(l2, label='Predicted')
      plt.xlabel('Data count - moving from Jan 2010 to Aug 2019')
      plt.ylabel('Carbon Dioxide (micromoles per mole)')
      plt.title('Plot of Actual and Predicted Carbon Dioxide at 21N/70.5E')
      plt.legend()
      plt.show()
```



This graph shows us the Predicted values vs. Actual values of Carbon Dioxide for the Training set.

Now, we move onto the quantitative analysis.



```
[38]: # Let us now evaluate our model quantitatively

[39]: # For this, we calculate R^2 (R-squared) and MSE (Mean Squared Error) for each model
      # R^2 is a measure of how close the data is to the fitted regression curve.
      # Higher the R^2, better is the model.
      # MSE is the mean of the squares of errors, i.e., the difference between the actual value and predicted value.
      # Lower the MSE, better is the model.

[40]: # For 21N/70.5E

      print("The R-squared = ",r1.score(X_new, Y1))
      print("MSE = ",mean_squared_error(Y1, r1_pred))

      The R-squared =  0.5491830830818207
      MSE =  23.769249285454624
```

What is R^2 ?

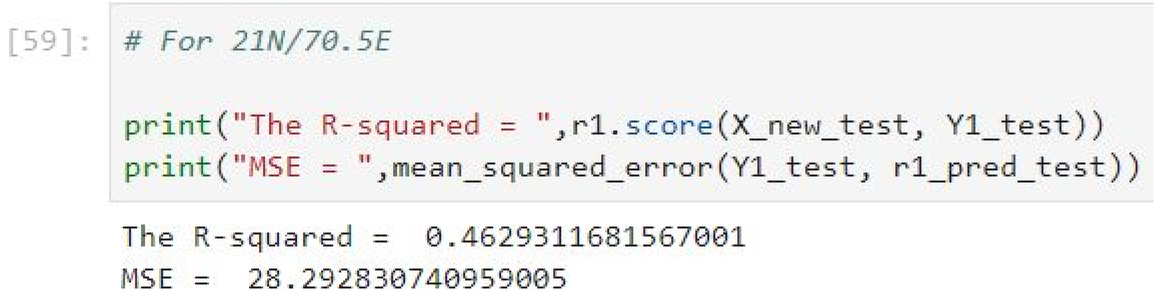
R^2 is a measure of how close the data is to the fitted regression curve. Higher the R^2, better is the model.

What is MSE?

MSE (Mean Square Error) is the mean of the squares of errors, i.e., the difference between the actual value and predicted value. Lower the MSE, better is the model.

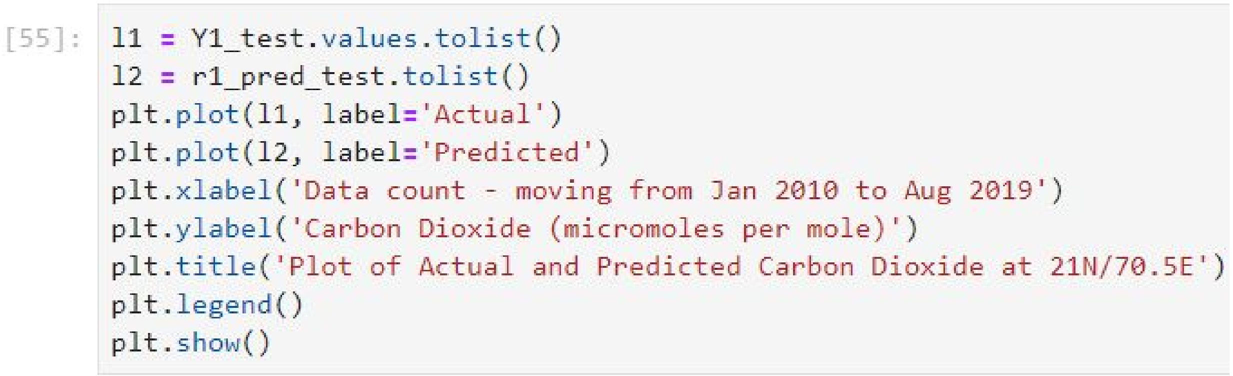
To test our model, we will use the data available from Jan 2010 to August 2019. The frequency of data is 32 days. The testing set contains data of 16 days after each day in the training set. For example, if January 9, 2010, is in the Training set, then January 25, 2010, is in the Testing set.

Following a similar process as before, we predict the data.

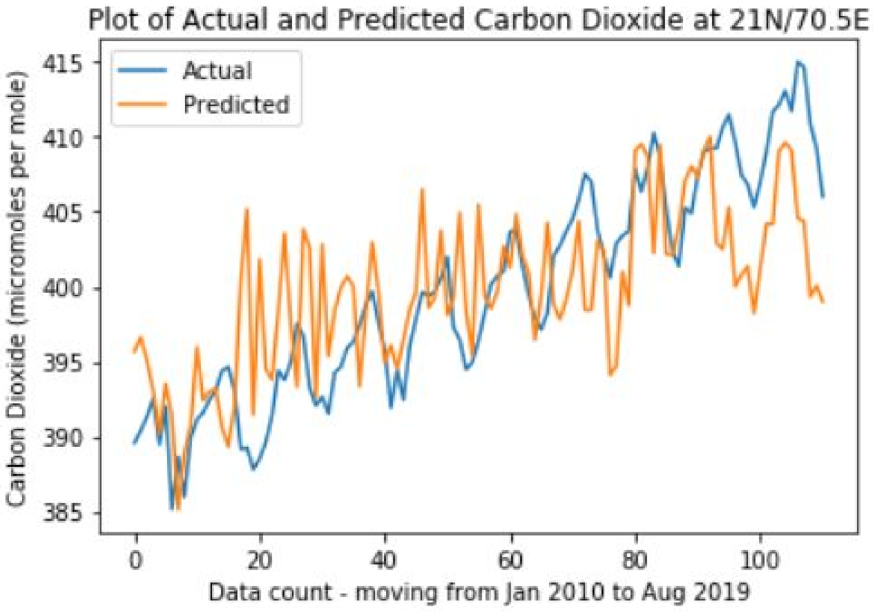


```
[59]: # For 21N/70.5E

      print("The R-squared = ",r1.score(X_new_test, Y1_test))
      print("MSE = ",mean_squared_error(Y1_test, r1_pred_test))
```

```
The R-squared =  0.4629311681567001
MSE =  28.292830740959005
```



```
[55]: l1 = Y1_test.values.tolist()
      l2 = r1_pred_test.tolist()
      plt.plot(l1, label='Actual')
      plt.plot(l2, label='Predicted')
      plt.xlabel('Data count - moving from Jan 2010 to Aug 2019')
      plt.ylabel('Carbon Dioxide (micromoles per mole)')
      plt.title('Plot of Actual and Predicted Carbon Dioxide at 21N/70.5E')
      plt.legend()
      plt.show()
```



We see that our model does a pretty job of predicting the Carbon Dioxide data, given NDVI and EVI values.

We can further increase the degree and create new models but, we observe that for our dataset, this leads to overfitting. Overfitting occurs when our model does very well in the training set but very bad in the test set. To avoid this, we do not increase the degree.

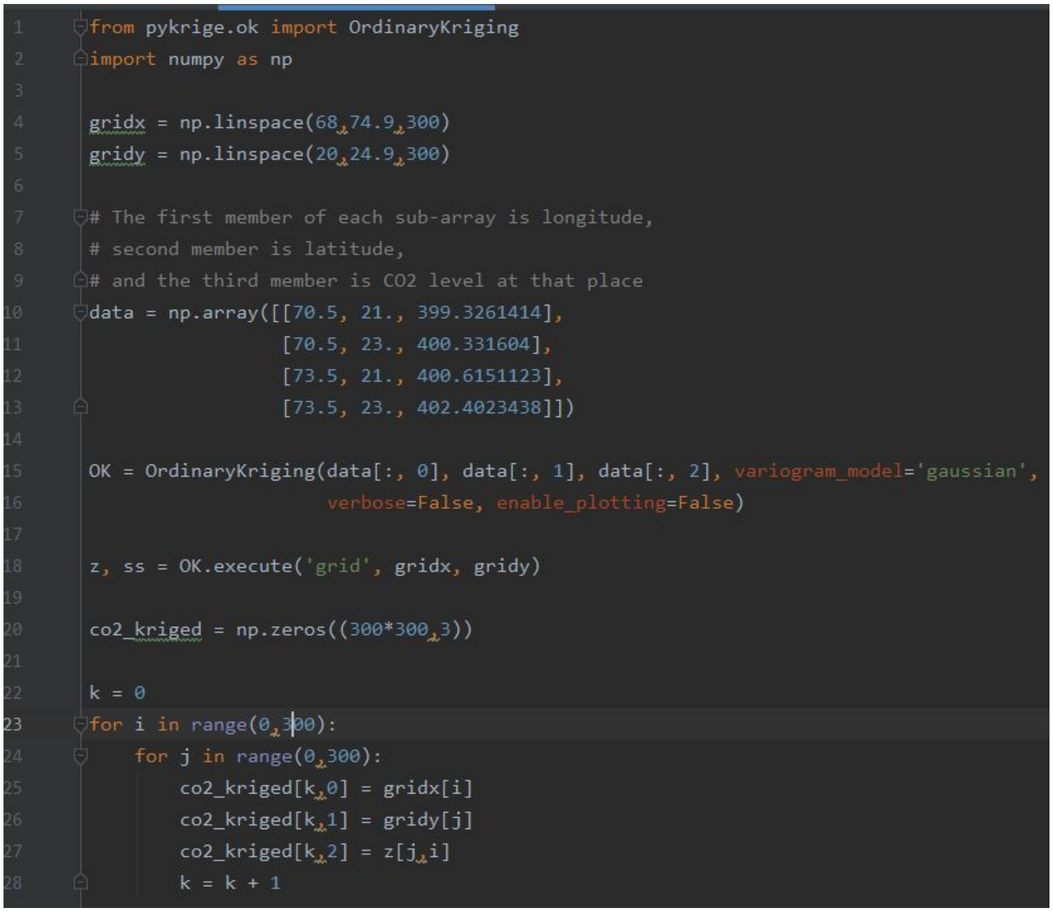
Kriging

Kriging is a method of interpolation for which the interpolated values are modeled by a Gaussian process governed by prior covariances. Under suitable assumptions on the priors, kriging gives the best linear unbiased prediction of the intermediate values.

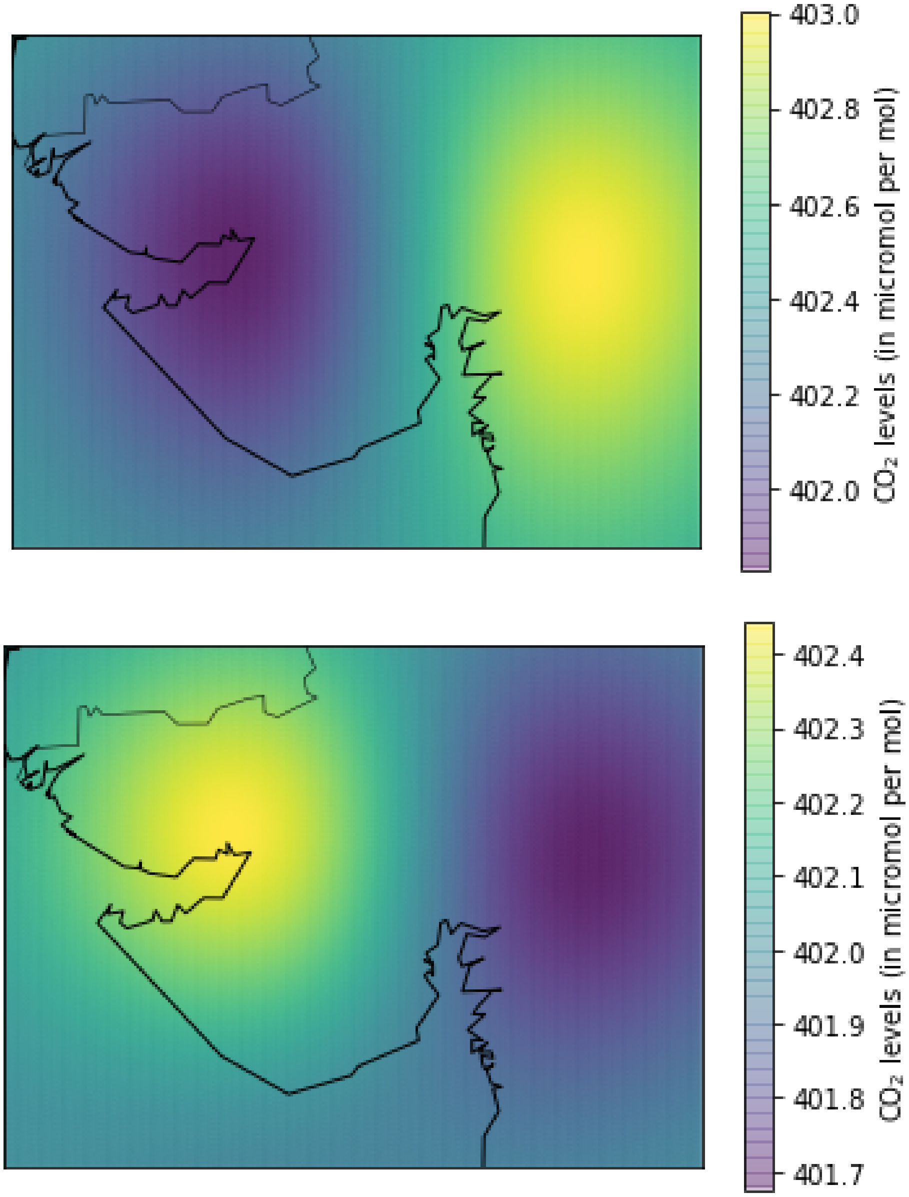
We used kriging to determine the values of CO_2 at other locations (other than the 4 given coordinates) whose data was not predicted by our machine learning model. We used a Gaussian model for the same.

We used kriging here because CO_2 values do not change very erratically. So, with values CO_2 levels at 4 coordinates in Gujarat, we predicted values of other 89996 coordinates.

The code for kriging is shown in the picture below:



```
from pykrige.ok import OrdinaryKriging
import numpy as np

gridx = np.linspace(68,74.9,300)
gridy = np.linspace(20,24.9,300)

# The first member of each sub-array is longitude,
# second member is latitude,
# and the third member is CO2 level at that place
data = np.array([[70.5, 21., 399.3261414],
                 [70.5, 23., 400.331604],
                 [73.5, 21., 400.6151123],
                 [73.5, 23., 402.4023438]])

OK = OrdinaryKriging(data[:, 0], data[:, 1], data[:, 2], variogram_model='gaussian',
                     verbose=False, enable_plotting=False)

z, ss = OK.execute('grid', gridx, gridy)

co2_kriged = np.zeros((300*300,3))

k = 0
for i in range(0,300):
    for j in range(0,300):
        co2_kriged[k,0] = gridx[i]
        co2_kriged[k,1] = gridy[j]
        co2_kriged[k,2] = z[j,i]
        k = k + 1
```


403.0
402.8
402.6
402.4
402.2
402.0
CO_2 levels (in micromol per mol)
402.4
402.3
402.2
402.1
402.0
401.9
401.8
401.7
CO_2 levels (in micromol per mol)

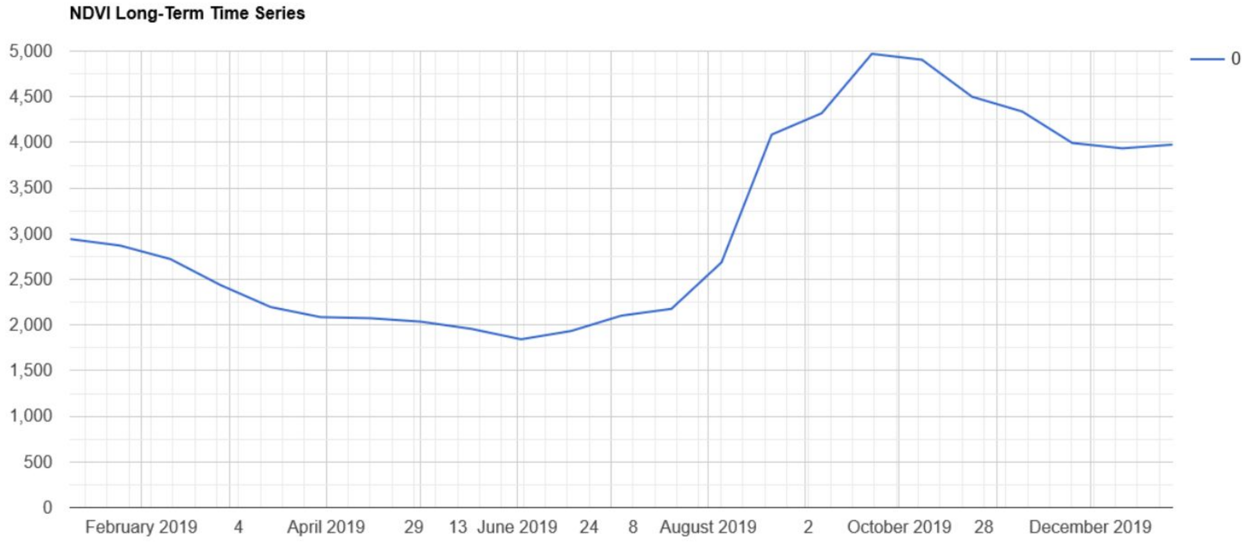
Above two images are CO2 data for two random days in Gujarat

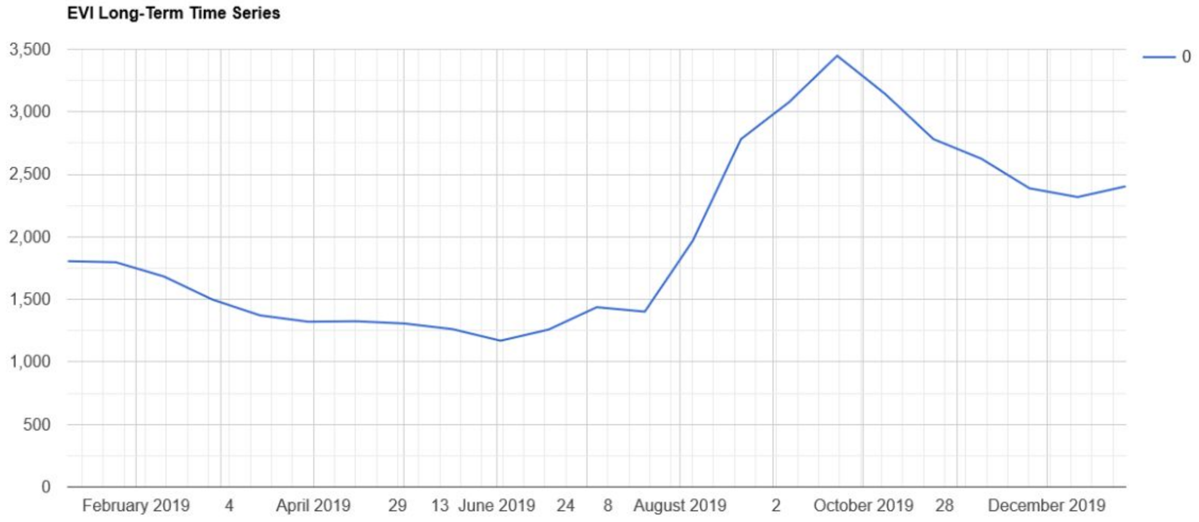
After obtaining the predicted data of carbon dioxide from our model for 4 coordinates (21N/70.5E, 23N/70.5E, 21N/73.5E, and 23N/73.5E), we use the Kriging method, to interpolate the values of carbon dioxide for the entire region of Gujarat.

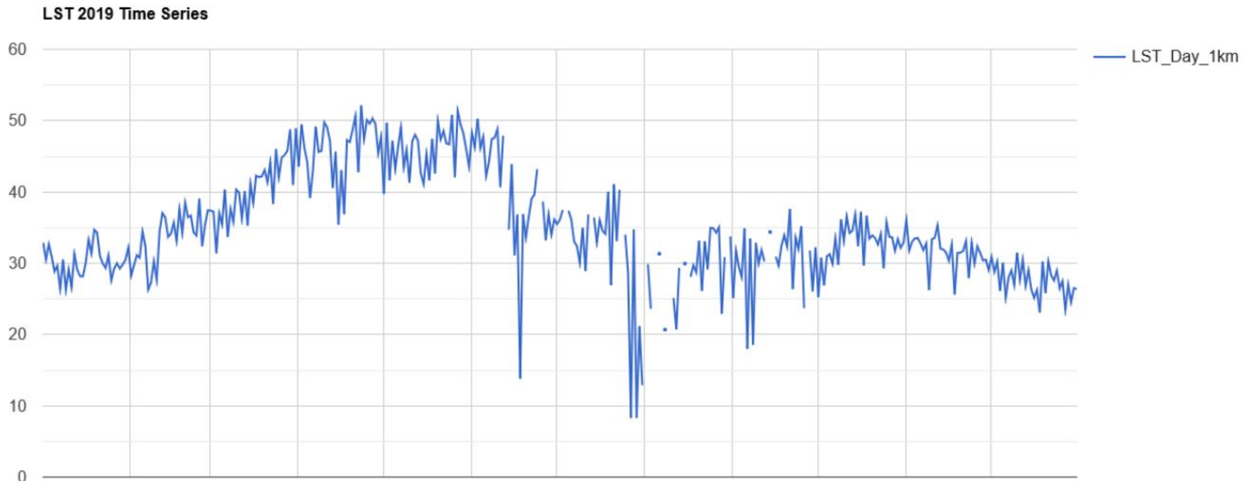
The obtained values, after Kriging, are plotted on the above maps. These maps represent the Carbon Dioxide value for the entire region of Gujarat.

Exploring Relationship Between NDVI & EVI with Temperature:

The following graphs are plots of values of NDVI, EVI, Precipitation, and Land Surface Temperature for the Year 2019:







As we observe and analyze the data, we can make the following observations and infer these conclusions: -

Observation :

As temperature increases, to some extent, NDVI and EVI also increase.

Conclusion :

When temperature increases, plants perform better photosynthesis. This eventually leads to increased consumption of carbon dioxide, reducing its amount in the atmosphere. Increased photosynthesis leads to better growth of vegetation, which in turn increases phenological factors, NDVI and EVI. Although, we must note that there is a saturation point of temperature, beyond which photosynthesis reduces.

Predicting Temperature with NDVI and EVI:

We develop a Polynomial Regression model to predict values of Land Surface Temperature on the basis of NDVI and EVI.

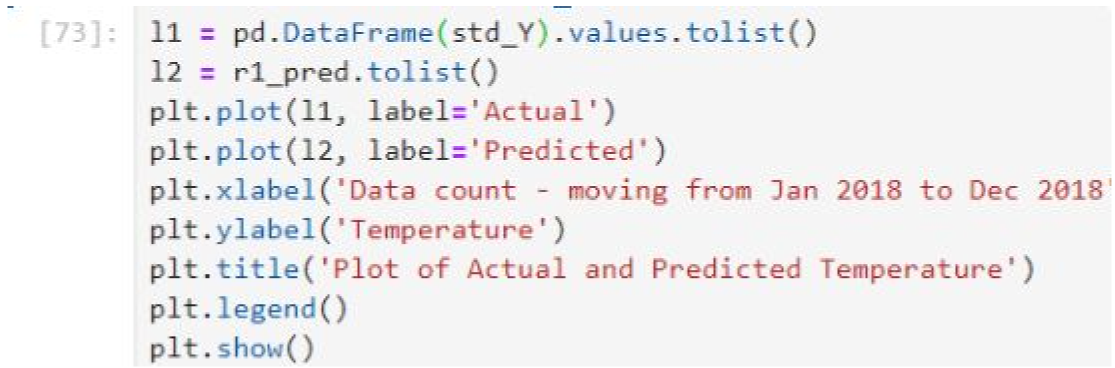
This time, we observe that a Polynomial Regression model with degree 5 gives the best results on scaling the data.

We use the data from 2018 for our training set. The frequency is set to 30 days. For testing purposes, we try to predict the values on March 14, 2019.

How was the data scaled?

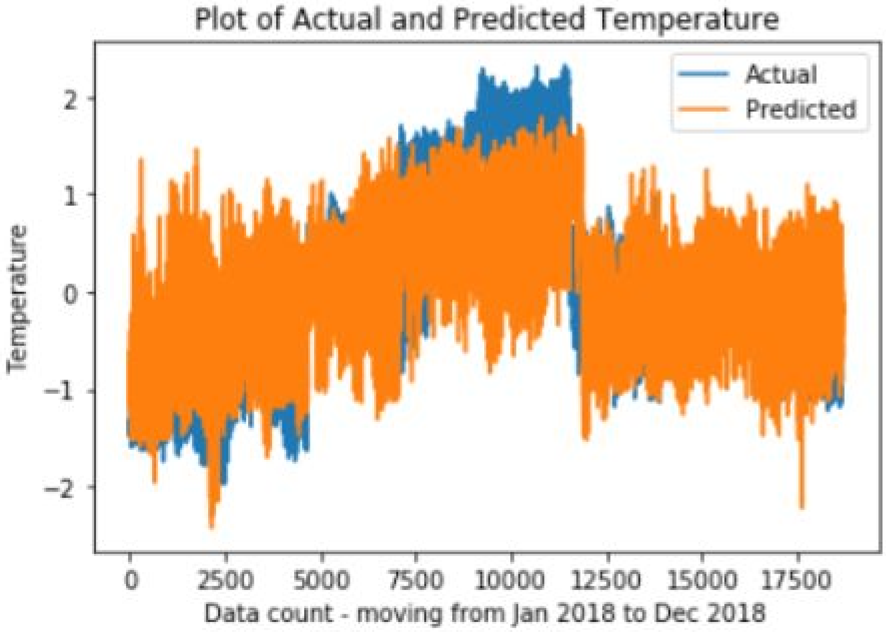
The data was scaled such that for every variable, the mean is 0 and the Standard Deviation (S.D.) is 1.

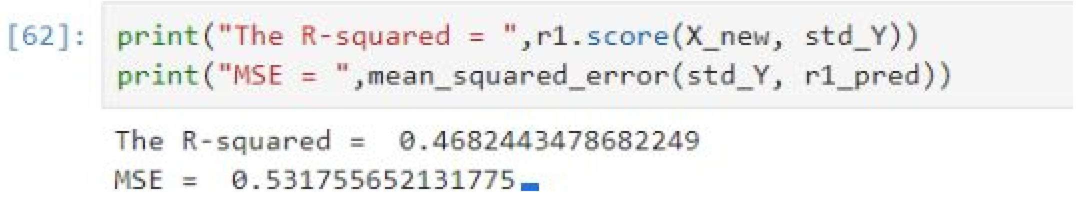
The following code is for plotting results of the model on training set itself:



```
[73]: l1 = pd.DataFrame(std_Y).values.tolist()
      l2 = r1_pred.tolist()
      plt.plot(l1, label='Actual')
      plt.plot(l2, label='Predicted')
      plt.xlabel('Data count - moving from Jan 2018 to Dec 2018'
      plt.ylabel('Temperature')
      plt.title('Plot of Actual and Predicted Temperature')
      plt.legend()
      plt.show()
```

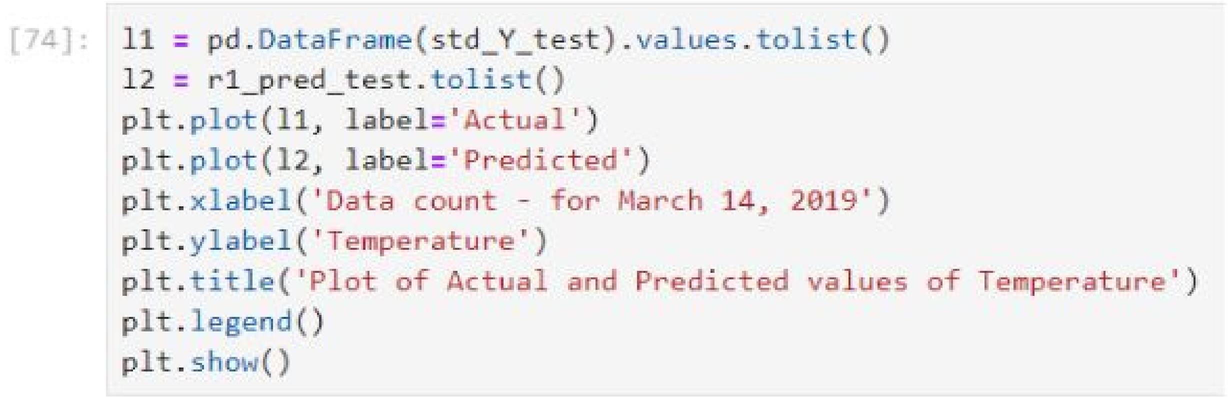
The following graphs and scores are for the training set:



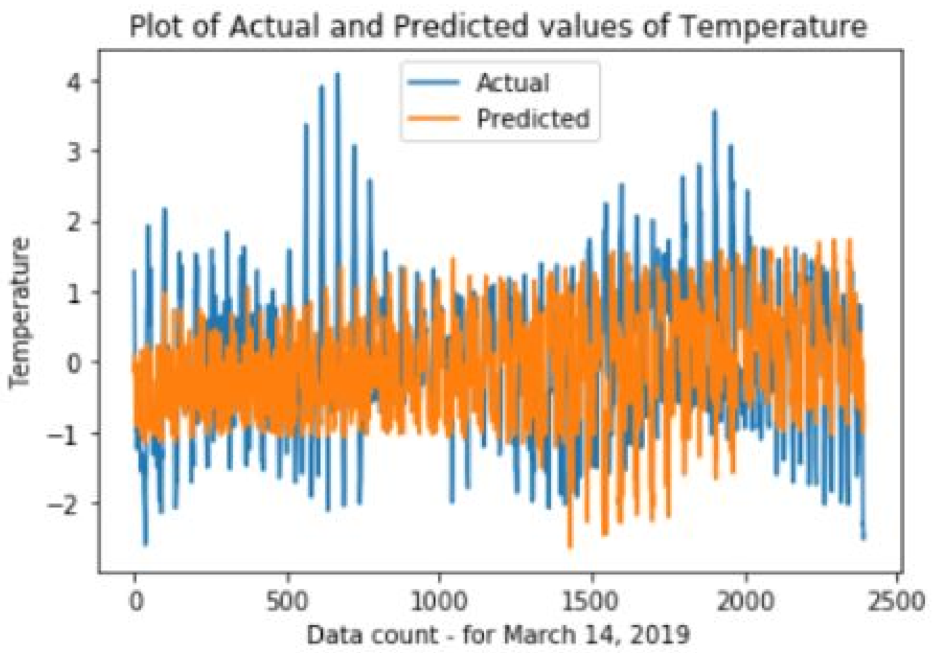


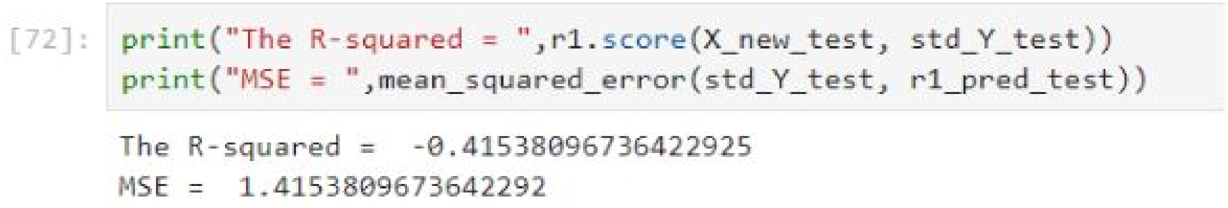
```
[62]: print("The R-squared = ",r1.score(X_new, std_Y))
      print("MSE = ",mean_squared_error(std_Y, r1_pred))

      The R-squared =  0.4682443478682249
      MSE =  0.531755652131775
```

The following graph and scores are for the testing set:

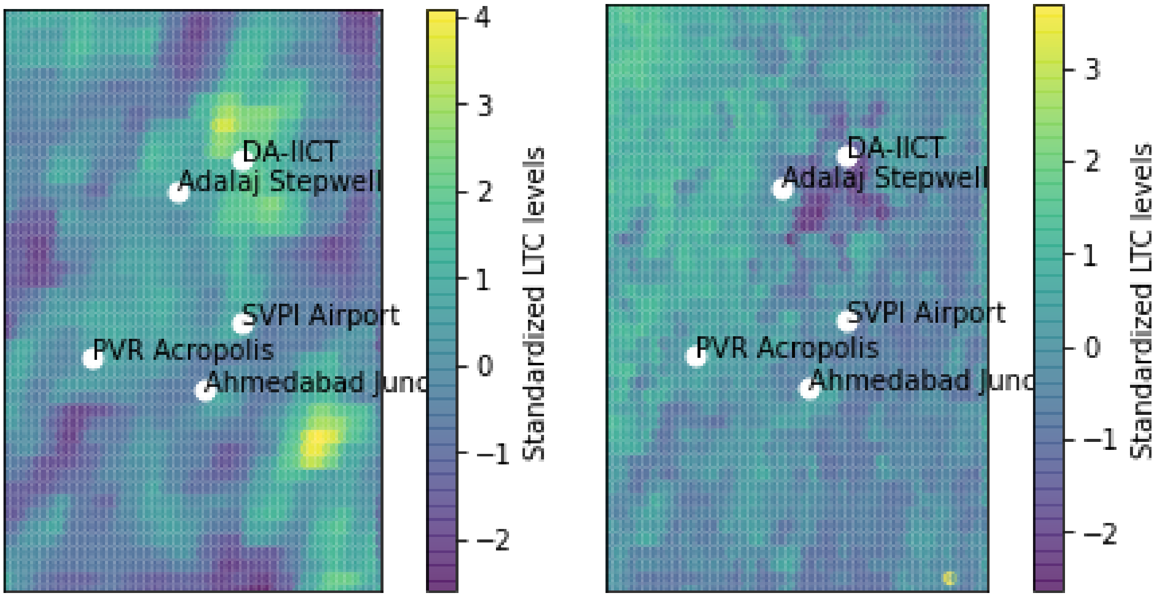


```
[74]: l1 = pd.DataFrame(std_Y_test).values.tolist()
      l2 = r1_pred_test.tolist()
      plt.plot(l1, label='Actual')
      plt.plot(l2, label='Predicted')
      plt.xlabel('Data count - for March 14, 2019')
      plt.ylabel('Temperature')
      plt.title('Plot of Actual and Predicted values of Temperature')
      plt.legend()
      plt.show()
```





```
[72]: print("The R-squared = ",r1.score(X_new_test, std_Y_test))
      print("MSE = ",mean_squared_error(std_Y_test, r1_pred_test))

      The R-squared =  -0.41538096736422925
      MSE =  1.4153809673642292
```



The above two images (left to right) show real and the predicted values of standardized LTC values, respectively

The test data consisted of data from 14 March 2019. From the above two images, we see some similarities between them. The portion to the right DA-IICT and SVPI Airport do resemble, and the portion below Ahmedabad junction is also a little bit similar. There are some discrepancies though (around DA-IICT, region to the left of the Adalaj Stepwell and on the lower right corner of the graph).

We can conclude that we won't be able to predict exact LTC(Land Surface Temperature) values just with the NDVI values, but we can surely get closer.

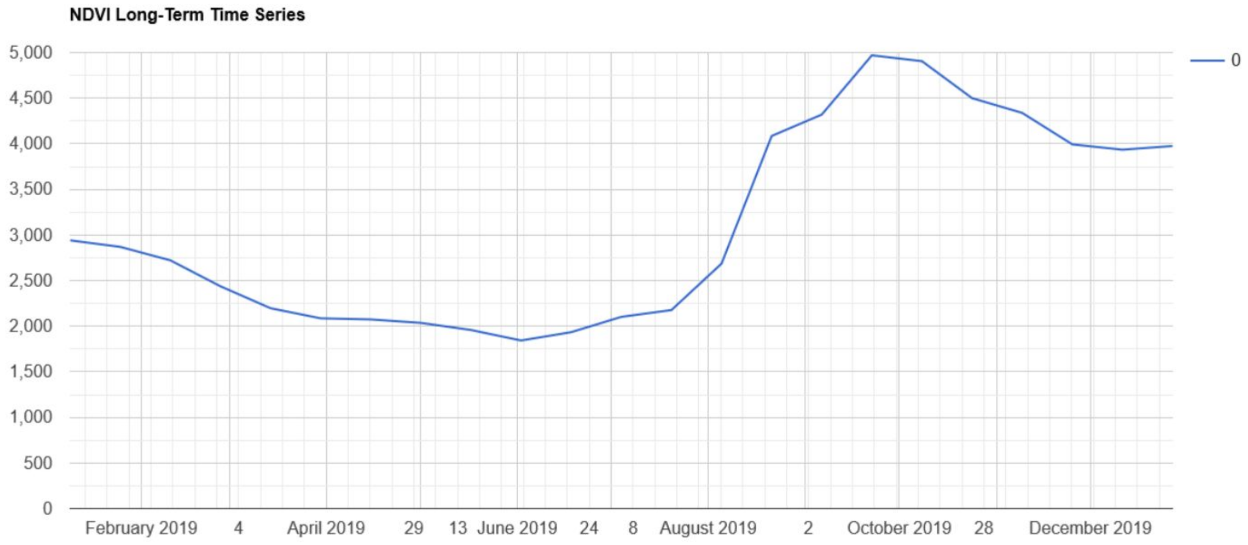
(Note:- We see that the value of one scatter point on the graph is yellow. Actually it was manually done just so that we get almost similar scale on both the maps.)

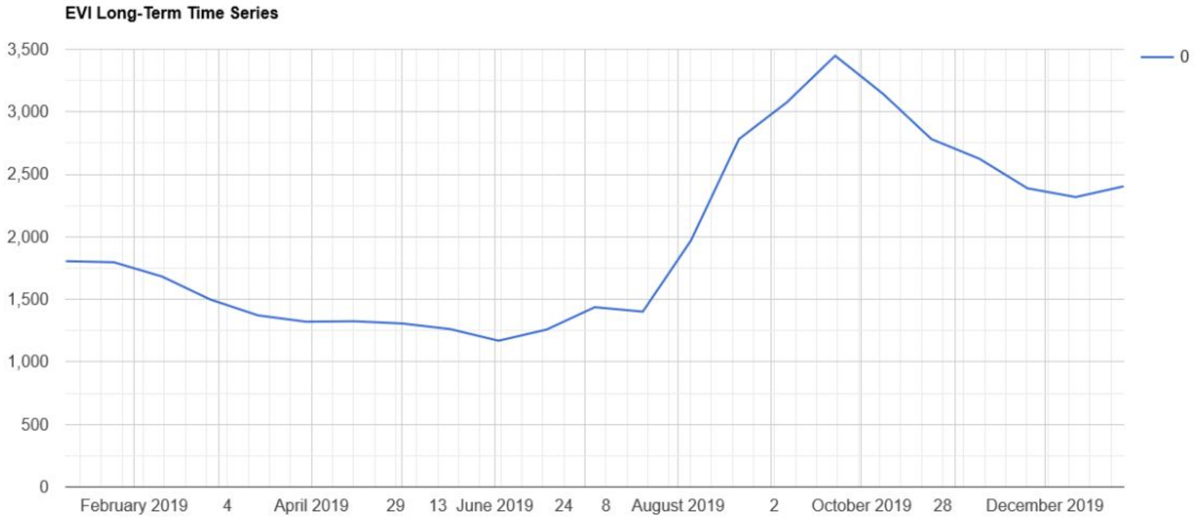
We can see that our model has some problems in predicting higher values of LTC. One of the reasons might be that NDVI shows the plant health, and photosynthesis occurs best at one optimal temperature and as we go above and below that temperature the NDVI reduces. Therefore, when NDVI values are low then the algorithm has to decide whether the LTC values were higher or lower than the optimal temperature. This might cause some confusion for the algorithm (confusion as in that it will maybe always predict low LTC values).

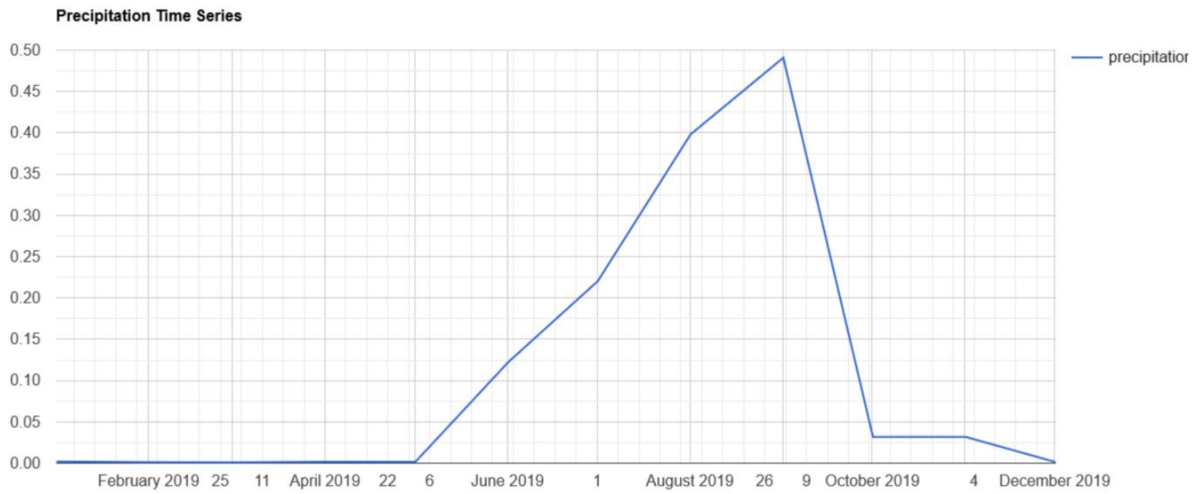
For plotting, we used the Basemap extension of matplotlib library of Python.

Analyzing the relationship between NDVI and EVI with Precipitation:

The following graphs are plots of values of NDVI, EVI and Precipitation for the Year 2019:







Observation :

NDVI, EVI, and Precipitation increase as the months July/ August approach. They continue to rise until October. This period, in India, is known as the monsoon season.

Conclusion :

NDVI and EVI, the phenological factors, denote the flourish of vegetation. In the monsoon season, when precipitation is high, plants are well-irrigated and their growth increases. This leads to an increase in the values of NDVI and EVI.

Citations/Articles

Determination of Vegetation Thresholds for Assessing Land Use and Land Use Changes in Cambodia using the Google Earth Engine Cloud-Computing Platform

A dataset of 30m annual vegetation phenology indicators (1985–2015) in urban areas of the conterminous United States

Phenology and its role in carbon dioxide exchange processes in northern peatlands

Monitoring vegetation phenology using MODIS

Climate change and its effects on vegetation phenology across ecoregions of Ethiopia

https://developers.google.com/earth-engine/playground

https://modis.gsfc.nasa.gov/data/

https://developers.google.com/earth-engine/datasets/catalog/MODIS_006_MYD13A1

https://www.researchgate.net/publication/225677736_Spatial_patterns_of_vegetation_phenology_metrics_and_related_climatic_controls_of_eight_contrasting_forest_types_in_India_-_Analysis_from_remote_sensing_datasets/citation/download

The Phenology Handbook by Brian P Haggerty and Susan J Mazer, University of California, Santa Barbara

https://budburst.org/phenology-defined

https://gisgeography.com/ndvi-normalized-difference-vegetation-index/

https://earthobservatory.nasa.gov/features/MeasuringVegetation/measuring_vegetation_2.php

https://www.vineview.com/evi-vs-ndvi-whats-difference/