|  |  |
| --- | --- |
| A picture of a winding road and trees  Telangana Forest Analysis  Using Data Cubes | Abstract  Unleash the Power of Geospatial Analysis: Explore, Analyze, and Visualize Satellite Imagery with Ease  Karnati Sai Krishna  Project School Documentation |

Project Report

**Project Title:** Telangana Forest Analysis using Data Cubes

**Project Members:**

K. Sai Krishna - 21BD1A050K

M. Venkata Varsha - 21BD1A050X

M. Sri Tharun - 21BD1A050V

N. Sumanth - 21BD1A0515

T. Jayadeep - 21BD1A051W

**Mentor:** Ram Mohan

**Domain:** Remote sensing

**Objective:** Develop a datacube using satellite images for analysis.

**Technologies:** Open Data Cube, QGIS, Flask, Machine Learning Algorithms

**Output:** The expected final product is a functional platform that can provide users with accurate and timely information about various changes in a region.

Contents

[1 Objectives 4](#_Toc140515066)

[2 Project Overview 5](#_Toc140515067)

[3 Setting up the Data Cube 7](#_Toc140515068)

[3.1 Data Cube Setup in Ubuntu Operating System 7](#_Toc140515069)

[3.2 Data Cube Setup in Docker Container 8](#_Toc140515070)

[4 Satellite Images and their Analysis 9](#_Toc140515071)

[4.1 Sentinel-2 9](#_Toc140515072)

[4.2 Landsat: 9](#_Toc140515073)

[5 Analysis through QGIS 11](#_Toc140515074)

[5.1 Installing QGIS 11](#_Toc140515075)

[5.2 Load Satellite Images 11](#_Toc140515076)

[5.3 Image Analysis 11](#_Toc140515077)

[5.4 Visualization and Output 13](#_Toc140515078)

[6 Analysis in Jupyter Notebook 14](#_Toc140515079)

[6.1 Install Required Libraries 14](#_Toc140515080)

[6.2 Load Satellite Data 14](#_Toc140515081)

[6.3 Preprocess the Data 15](#_Toc140515082)

[6.4 Visualize Results and Changes 15](#_Toc140515083)

[7 Web Application 17](#_Toc140515084)

[7.1 Frontend – User Interaction 17](#_Toc140515085)

[7.2 Backend – Flask API 18](#_Toc140515086)

[8 Results 19](#_Toc140515087)

[8.1 Analysis Results 19](#_Toc140515088)

[8.2 Visualizations 20](#_Toc140515089)

[8.3 Data Interpretation 21](#_Toc140515090)

[8.4 User Experience 21](#_Toc140515091)

[8.5 Performance and Scalability 21](#_Toc140515092)

[9 Future Scope 22](#_Toc140515093)

[9.1 Additional Analysis Types 22](#_Toc140515094)

[9.2 Enhanced Visualization 22](#_Toc140515095)

[9.3 User-Driven Customization 22](#_Toc140515096)

[9.4 Performance Optimization 23](#_Toc140515097)

[9.5 Integration with External Data Sources 23](#_Toc140515098)

[9.6 User Authentication and Access Control 23](#_Toc140515099)

[10 References 25](#_Toc140515100)

# Objectives

Telangana is full of flora and fauna. In order to maintain its integrity, we need a proper tool to efficiently monitor the changes which are taking place in the forest areas. Our project is to develop a web application with a user interface that provides the detailed analysis of the forest cover in different regions and a monitoring tool for the vegetation in Telangana. The objectives of our project are:

**Forest change Detection and Monitoring:** An additional objective of the project is to develop tools and methodologies for timely forest change detection and monitoring using remote sensing data. By analysing the various bands data which is collected by satellites, the project aims to provide a clear change detection over a period of time by using various indices, assist in forest management efforts, and contribute to the reduction of deforestation in forest areas.

**Future Forest Cover Prediction:** The project also incorporates machine learning algorithm to predict forest cover change in the future. By leveraging historical satellite imagery data and other relevant variables, the project aims to develop models that can forecast changes in forest cover over time. This objective seeks to enhance our understanding of future forest dynamics, inform land-use planning decisions, and support proactive forest conservation strategies.

**Web Application Development:** In parallel with the remote sensing analysis, the project aims to develop a web application that offers proper analysis and visualisation of the selected forest area. The application will provide users with tools for interactive mapping, image processing, visualising forest change, and data visualisation. This objective aims to facilitate access to the analysed information, enhance decision-making processes, and support various people involved in forest management and conservation efforts.

# Project Overview

Our project aims to conduct a comprehensive analysis of forest dynamics using remote sensing data and machine learning techniques. The goal is to gain valuable insights into changes in forest area and understand the driving factors behind these changes. By leveraging the power of satellite imagery and advanced analytics, we aim to provide actionable information for effective forest management and conservation.

The project encompasses several key components. First, we acquire satellite imagery data from reliable sources, ensuring sufficient coverage over the study area. The data includes multispectral imagery capturing various spectral bands, enabling us to extract essential vegetation indices and environmental variables.

Next, we preprocess the data to remove noise, correct for atmospheric effects, and extract relevant features. Also, we calculate key vegetation indices such as NDVI, NDMI, and EVI, which serve as critical indicators of vegetation health, water content, and biomass.

With the pre-processed data and derived features in hand, we employ machine learning algorithms, specifically the Random Forest algorithm, to model forest dynamics. By training the model on historical data and incorporating environmental variables, we can predict and analyse changes in forest area over time. The trained model provides valuable insights into the relative importance of different factors in driving forest dynamics.

Throughout the project, we emphasize the importance of accuracy, reliability, and interpretability. We employ rigorous validation techniques to assess the performance of our models and ensure the robustness of our results. Furthermore, we focus on clear and transparent communication of our findings, providing actionable information for forest managers, policymakers, and conservation organizations.

Our project relies on the data collected by various satellites. To efficiently store and access the massive amount of data, we have implemented a cutting-edge solution known as a Data Cube. The Data Cube serves as a powerful storage system specifically designed for time series analysis, ensuring seamless access and retrieval of data for our project's needs. We have used Flask in Python as our backend system in order to provide an API for our application to communicate with the Data Cube. We have also used Open Data Cube as our API interface in python to access data from the Data Cube.

# Setting up the Data Cube

Setting up the Data Cube involves many steps which includes setting up a Python environment and installing necessary modules, getting the required satellite images, adding them up into the Data Cube, visualising them in Data Cube Explorer in order to conform their installation and presence of datasets.

Data Cube can be set up in two ways. One by directly setting up in an Ubuntu machine. Other, by setting it up using Cube in a box, which is a Docker based container setup. We can choose either of them, which doesn’t matter for the Data Cube final product.

## Data Cube Setup in Ubuntu Operating System

**Datacube** runs the best on Ubuntu version 20.04. There is also a third-party tested software called as OSGeoLive VM which is based on LUbuntu 22.04. This provides pre-configured applications for a range of geo-spatial use-cases that includes storage, publishing, viewing, analysis and manipulation of data.

The steps that involves in setting up a Data Cube are:

1. Setting up the Python Environment
2. Postgres Database configuration for testing
3. Data Cube installation
4. Verification of the installation
5. Creating a Database for the actual Data Cube
6. Adding Product Definitions
7. Adding Satellite imagery to the file system
8. Transforming Metadata to YAML
9. Add datasets to the Data Cube
10. Verify everything works

## Data Cube Setup in Docker Container

Data Cube can also be setup by using the pre-built cube-in-a-dash Docker Container. This can be found in the reference by Open Data Cube GitHub <https://github.com/opendatacube/cube-in-a-box>.

The steps involved in this are:

1. Build a local environment: docker-compose build
2. Start a local environment: docker-compose up
3. Set up your local Postgres database

# Satellite Images and their Analysis

Satellite images play a vital role in remote sensing projects by providing a comprehensive and objective view of the Earth's surface from a considerable distance. Remote sensing involves the collection and analysis of information about the Earth's features, environment, and phenomena without direct physical contact.

Earth observatory satellites are a category of satellites specifically designed to observe and monitor the Earth's surface, atmosphere, and various environmental phenomena. The satellites we considered for our project are Landsat and sentinel-2.

## Sentinel-2

The Sentinel-2 mission, part of the European Space Agency's (ESA) Copernicus program, consists of two identical satellites, Sentinel-2A and Sentinel-2B. These satellites provide high-resolution multispectral imagery, enabling detailed land cover mapping, agricultural monitoring, and environmental management.

Sentinel-2 captures data in 13 spectral bands. These bands include the Coastal Aerosol band (sensitive to suspended matter in water), Blue band (for water and vegetation reflectance), Green band (vegetation health), Red band (vegetation analysis and soil moisture), and several Red Edge bands (precise vegetation health assessment).

## Landsat:

Landsat satellites, operated by the U.S. Geological Survey (USGS) and NASA, have been capturing imagery of the Earth's surface since 1972. The Landsat program provides a long-term record of the planet's changes, monitoring land cover, vegetation health, urban growth, and other land-related phenomena. Landsat 8 and Landsat 9 are the current active satellites in the series.

Landsat satellites also capture data in multiple spectral bands. These include visible, near-infrared, short-wave infrared, and thermal infrared bands. Each band represents specific characteristics such as land cover, vegetation health, water bodies, and thermal information.

# Analysis through QGIS

QGIS is a powerful open-source geographic information system (GIS) software that supports various remote sensing capabilities. By following the instructions below, you will be able to load satellite images, perform basic preprocessing, conduct analysis, and visualize the results using QGIS.

The steps generally involved in analysis using QGIS are:

## Installing QGIS

If you haven't installed QGIS yet, visit the official QGIS website (https://qgis.org) and download the appropriate version for your operating system. Follow the installation instructions to set up QGIS on your computer.

## Load Satellite Images

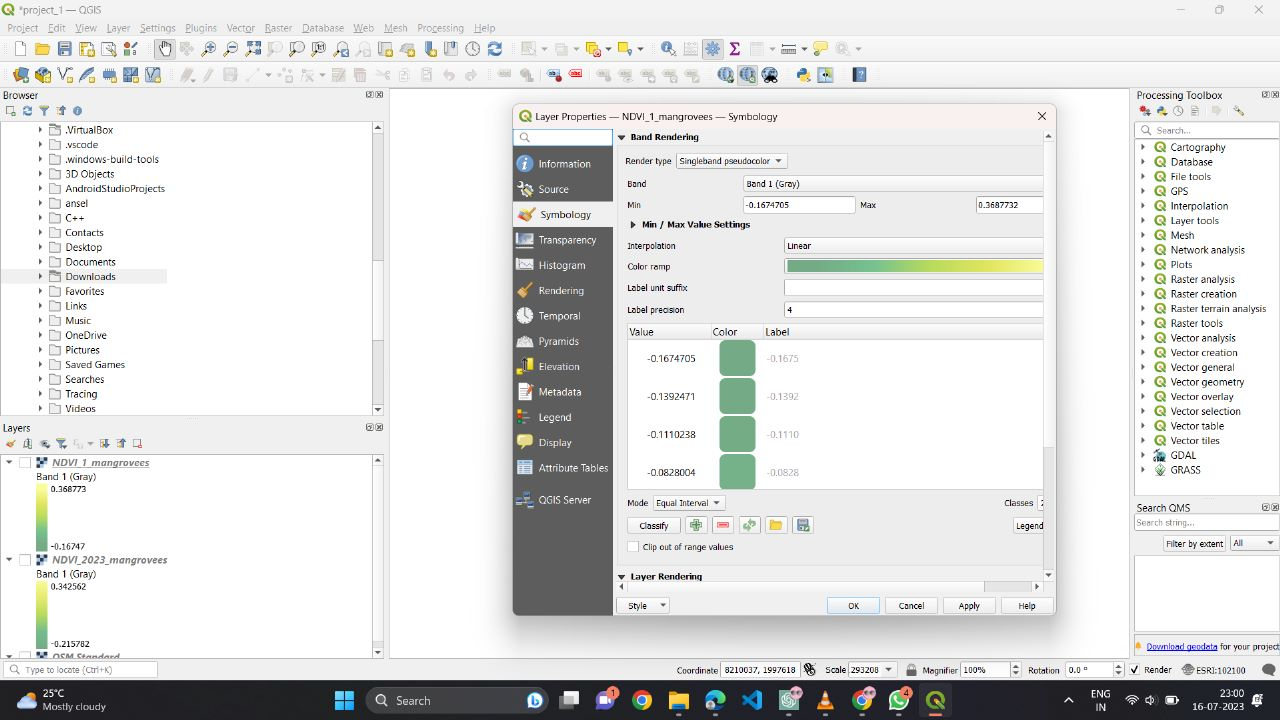
Launch QGIS. In the menu bar, click on "Layer" and select "Add Layer" > "Add Raster Layer."

Browse and select the satellite image file you want to analyze. QGIS supports various satellite image formats, including GeoTIFF, JPEG, and PNG.

Click "Open" to load the satellite image into the QGIS canvas.

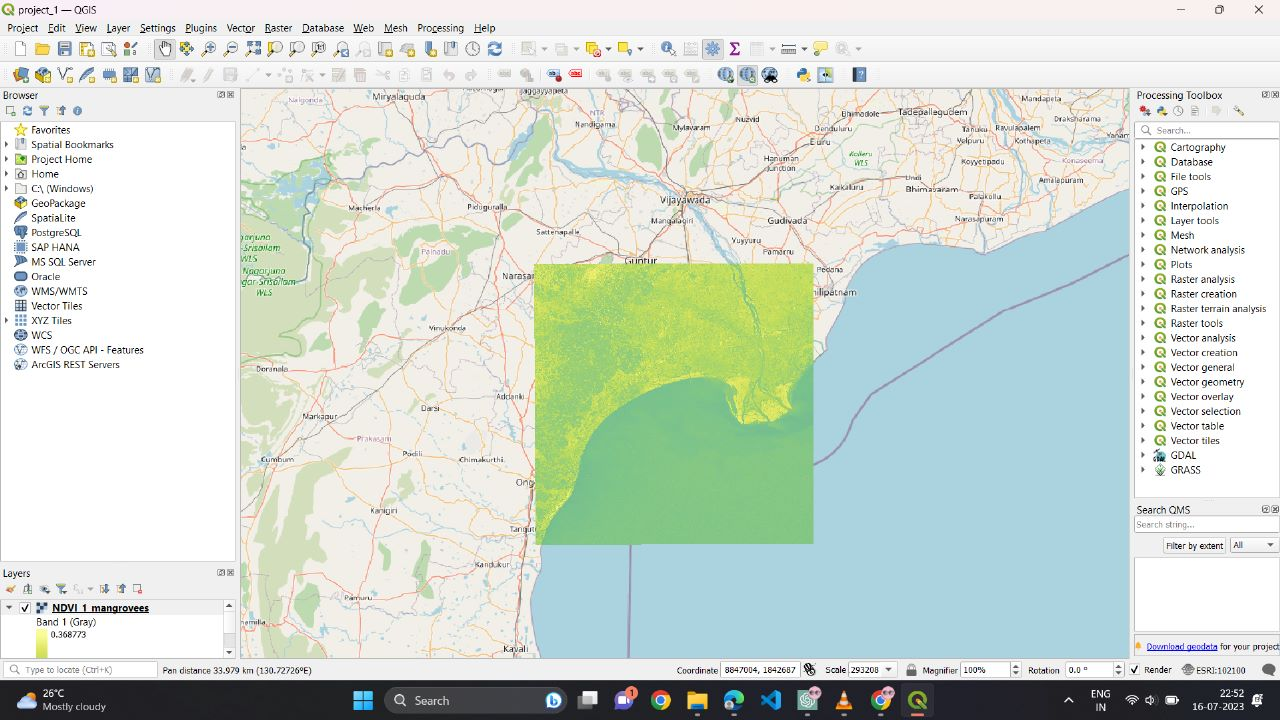
## Image Analysis

QGIS offers a wide range of tools and plugins for image analysis. Here are some commonly used tools and techniques:

1. Band Combinations: Create composite images by combining different bands to enhance certain features. Use the "Raster" > "Miscellaneous" > "Build Virtual Raster" tool to create a new composite image.
2. Image Classification: Perform image classification to categorise land cover types. QGIS provides plugins like Semi-Automatic Classification Plugin (SCP) or Orfeo Toolbox for image classification.
3. Indices Calculation: Calculate spectral indices like NDVI (Normalised Difference Vegetation Index) or NDBI (Normalised Difference Built-Up Index) to analyse vegetation or built-up areas. Use the "Raster Calculator" tool in the Processing Toolbox to perform these calculations.
4. Change Detection: Compare two or more satellite images to identify changes over time. The "Change Detection" plugin in QGIS helps in visualising and quantifying these changes.
5. Image Enhancement: Apply image enhancement techniques like contrast adjustment, histogram equalisation, or pansharpening to improve the visual quality of satellite images. These tools are available in the "Raster" menu.

## Visualization and Output

QGIS provides various visualization techniques to display and interpret the analysis results. Here are some visualization options:

* 1. Layer Styling: Adjust the symbology of the satellite image layer to visualise different bands or highlight specific features. Use the "Layer Styling" panel to modify colours, transparency, or blending modes.
  2. Add Additional Layers: Overlay vector data, such as shapefiles or point data, on top of the satellite image to provide context or perform spatial analysis.
  3. Generate Output: Export your analysis results or maps to different formats, such as PDF, image formats (PNG, JPEG), or geospatial formats (GeoTIFF, shapefile).

QGIS provides a comprehensive platform for analyzing satellite images and offers a wide range of functionalities to support remote sensing projects.

# Analysis in Jupyter Notebook

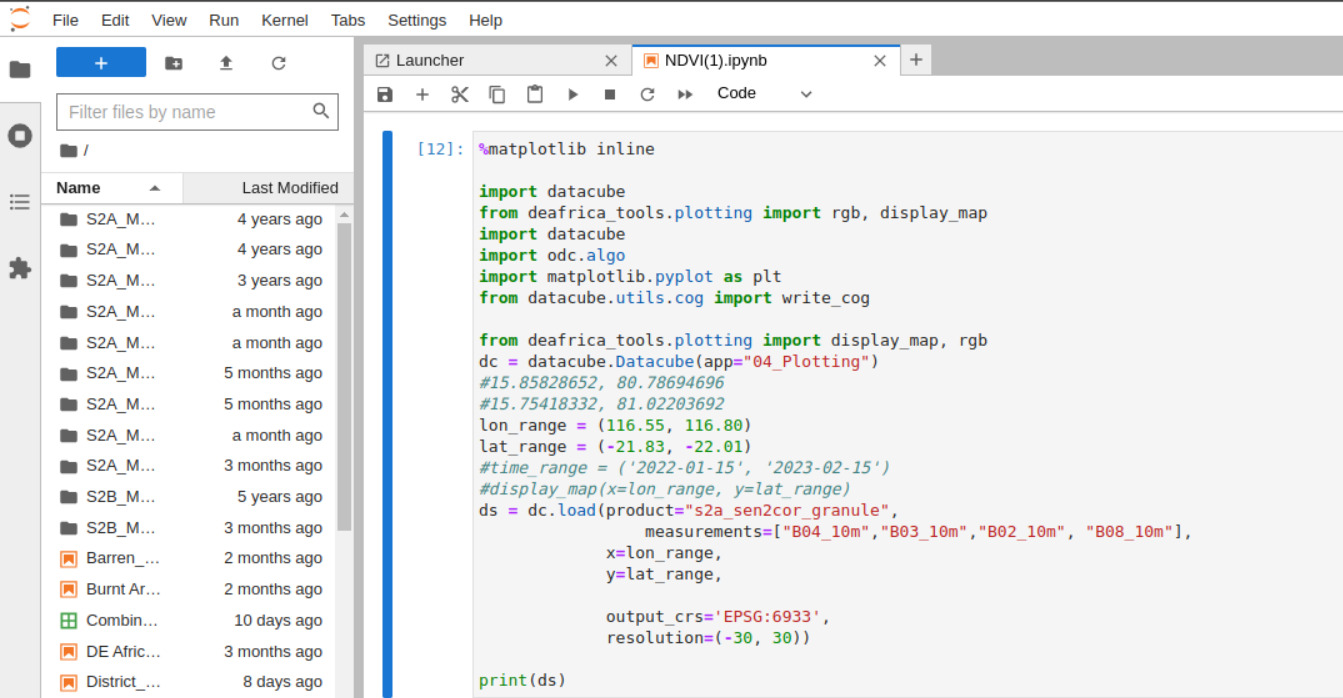
Analyzing satellite images using Jupyter Lab, Data Cube, Xarray, and Matplotlib. By following the instructions below, you will be able to extract satellite data from the Data Cube, process it using Xarray, and visualize vegetation changes using Matplotlib in a Jupyter Lab environment.

The following steps are to be followed in order to get the analysis using a Jupyter Notebook:

## Install Required Libraries

Ensure that you have Jupyter Lab, Data Cube, Xarray, and Matplotlib installed in your virtual environment. Import the necessary libraries in your Jupyter Notebook. Set up the Data Cube configuration to connect to your data source.

## Load Satellite Data

Specify the area of interest using coordinates or a bounding box and define the time range for analysis. Load the satellite data from the Data Cube using the APIs provided by the datacube module.

## Preprocess the Data

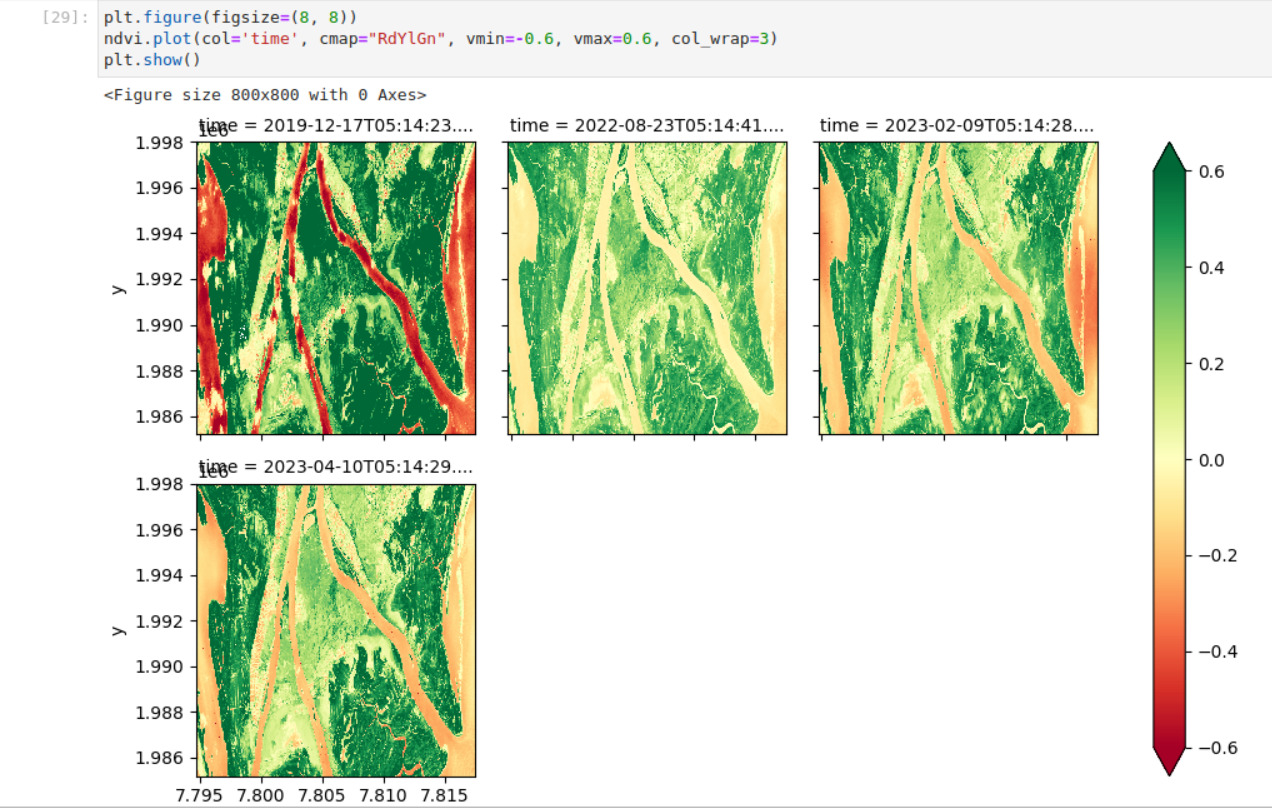
Preprocess the dataset as required. For vegetation analysis, you can calculate the Normalized Difference Vegetation Index (NDVI) using the data extracted from the Data Cube in the form of an Xarray.

ndvi = (dataset["nir"] - dataset["red"]) / (dataset["nir"]+dataset["red"])

Perform additional preprocessing steps such as resampling, filtering, or masking, as per your analysis requirements.

## Visualize Results and Changes

Plot the vegetation change using Matplotlib image plot or plot provided by the datacube.



In this way, satellite data from the Data Cube can be extracted using the Data Cube APIs, processed it using Xarray functions, and vegetation changes can be visualized (for example) using Matplotlib in a Jupyter Lab environment. This workflow allows you to perform detailed satellite image analysis and gain insights into vegetation dynamics based on the specified coordinates area.

# Web Application

The web application involves in combination of several individual components such as the frontend, the backend and the Data Cube already set up using Postgres. This, hence, needs to be done in an efficient way such that we provide the best experience to the user and also reducing the efforts and load on our systems.

The major components involved in the Web Application of this project are as follows:

## Frontend – User Interaction

The frontend part of the application is built by using the frontend technologies HTML, CSS, and JavaScript. Our requirements of the frontend frameworks are getting done by these three languages.

The User is expected to select an Area of Interest in the provided map. The User is then expected to enter the filter parameters such as start and end date for their query and type of analysis they wish to perform.

The map used in this application is integrated by using Leaflet JS library, which is being provided by Open Street Map. The application utilised the base map provided by the Open Street Map and then Leaflet JS allows us to add layers on top of the base map, in various formats of shapes, such as rectangle and polygon.

We also use Leaflet JS to indicate the bounding boxes of our available datasets in the Data Cube in order to mention user to select the Area of Interest from the indicated regions only. This helps us to maintain a decent User Experience.

The app connects with the backend through the APIs we define in the backend by sending asynchronous requests to them. The results are shown in the frontend using visualisation tools such as Plotly JS, Chart JS and Matplotlib plot.

## Backend – Flask API

The backend of the web application is built using Flask, a Python web framework. The Flask API handles incoming requests from the frontend, performs data processing and analysis, and provides responses accordingly. It interacts with the Data Cube with the provided APIs, which is already set up using Postgres, to retrieve relevant geospatial data based on the user's selections and filters.

The Flask API defines endpoints that are responsible for handling various requests from the frontend, such as retrieving available datasets, performing analysis based on the selected AOI and analysis type, and returning the results to the frontend. It utilizes various libraries and tools such as Data Cube, Postgres, scikit-learn, Matplotlib, and geopy for data processing, analysis, and visualization.

The Flask API ensures efficient handling of requests by optimizing the data retrieval and processing operations, thereby providing a smooth User Experience and reducing the load on the system.

# Results

The web application provides users with a range of analysis types and presents the results in a visually appealing and interactive manner. Through the application's user-friendly interface, users can explore and interpret the analysis outcomes effortlessly. Here are the key results and features of the web application:

## Analysis Results

1. NDVI Analysis:
   * The Normalized Difference Vegetation Index (NDVI) analysis enables users to assess vegetation health and density.
   * The application generates NDVI maps that visualize the distribution of vegetation in the selected area of interest.
   * The NDVI values range from -1 to 1, with higher values indicating healthier and denser vegetation.
2. NDWI Analysis:
   * The Normalized Difference Water Index (NDWI) analysis aids in identifying water bodies and assessing water content.
   * NDWI maps are generated, providing insights into the presence and distribution of water in the selected area.
   * The NDWI values range from -1 to 1, with higher values indicating a higher concentration of water.
3. EVI Analysis:
   * The Enhanced Vegetation Index (EVI) analysis offers an advanced approach to evaluate vegetation health and density.
   * EVI maps are generated, providing a more accurate representation of vegetation conditions, especially in areas with high levels of aerosol content.
   * The EVI values range from -1 to 1, with higher values indicating healthier and denser vegetation.
4. Random Forest Analysis:
   * The Random Forest analysis utilizes machine learning techniques to predict forest cover and changes over time.
   * The application performs regression analysis based on the selected area and time range to estimate forest cover and future trends.
   * Interactive charts and graphs are generated using Plotly to illustrate the predicted forest cover and provide insights into future forest dynamics.

## Visualizations

* The web application leverages interactive maps powered by Leaflet JS to display the results of the analysis.
* Users can navigate the maps, zoom in/out, and explore the analysis outcomes at different spatial scales.
* The application also includes interactive charts and graphs created with Plotly, allowing users to visualize and analyse data trends in an intuitive manner.

## Data Interpretation

* The analysis results provide valuable information on vegetation health, water content, and forest dynamics in the selected area.
* Users can interpret the analysis outcomes by observing patterns, trends, and spatial variations in the displayed maps and visualizations.
* It is important to consider the context and any specific domain knowledge when interpreting the results and drawing meaningful insights.

## User Experience

* The web application offers a seamless and user-friendly experience for interacting with the analysis results.
* Users can easily select the area of interest, specify analysis parameters, and visualize the outcomes with minimal effort.
* The intuitive interface, combined with interactive elements, enhances user engagement and understanding of the analysis results.

## Performance and Scalability

* The web application is designed to handle large datasets efficiently, ensuring optimal performance even with complex analysis tasks.
* Backend optimizations, such as data indexing and caching, enable fast data retrieval and processing.
* The system architecture allows for scalability, accommodating future growth and accommodating increased user demand.

# Future Scope

The web application lays a strong foundation for further enhancements and expansion of its capabilities. Here are some potential areas of future development and improvements:

## Additional Analysis Types

* Expand the range of analysis types offered by the application to provide users with more insights into geospatial data.
* Integrate advanced analysis algorithms and techniques for specialized applications, such as land cover classification, change detection, or anomaly detection.
* Incorporate user feedback and domain-specific requirements to identify new analysis types that would benefit the target user base.

## Enhanced Visualization

* Continuously improve and refine the visualizations presented to users for a more informative and engaging experience.
* Explore advanced visualization techniques, such as 3D mapping, heatmaps, or time series animations, to convey complex geospatial information effectively.
* Integrate additional visualization libraries or tools to offer more customization options and flexibility to users in visualizing analysis results.

## User-Driven Customization

* Provide users with more control and customization options in selecting analysis parameters, visualizations, and data presentation.
* Allow users to save and share their analysis configurations and results for collaboration or future reference.
* Implement user feedback mechanisms to gather insights and suggestions for further improvements and customization options.

## Performance Optimization

* Continuously optimize the backend infrastructure and algorithms to handle larger datasets and complex analysis tasks more efficiently.
* Implement parallel processing techniques or distributed computing frameworks to speed up data processing and analysis.
* Explore cloud-based solutions for scalable and elastic resource allocation to handle increasing user demand and accommodate future growth.

## Integration with External Data Sources

* Enable integration with external geospatial data sources, such as satellite data providers or public APIs, to expand the range of available datasets for analysis.
* Incorporate real-time data feeds or sensor data for near-real-time analysis and monitoring of dynamic phenomena.
* Establish partnerships or collaborations to access specialized datasets that cater to specific user requirements or niche applications.

## User Authentication and Access Control

* Implement user authentication and access control mechanisms to ensure data security and privacy.
* Allow users to save and manage their analysis projects, datasets, and preferences securely.
* Provide role-based access control to facilitate collaboration among multiple users with varying levels of permissions and privileges.

# References

* <https://deafrica.africageoportal.com>
* <https://docs.digitalearthafrica.org/en/latest/sandbox/notebooks/Beginners_guide/02_Products_and_measurements.html>
* <https://sandbox.digitalearth.africa/hub/login?next=%2Fhub%2F>
* <https://www.dea.ga.gov.au/>
* <https://www.sentinel-hub.com/develop/custom-scripts/>
* <https://scihub.copernicus.eu/dhus/#/home>
* <https://www.opendatacube.org/>
* <https://datacube-core.readthedocs.io/en/latest/about-core-concepts/index.html>
* <https://opensourceoptions.com/blog/remote-sensing-with-qgis-calculate-ndvi/>
* <https://earthexplorer.usgs.gov/>
* <https://docs.dea.ga.gov.au/notebooks/Beginners_guide/08_Intro_to_xarray.html>
* <https://flask.palletsprojects.com/en/2.3.x/>