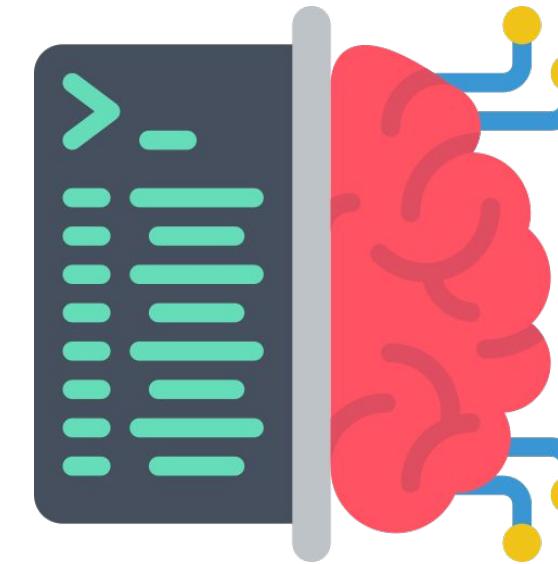


Introduction to **Google Earth Engine**

Python API for Machine Learning based
classification



Mirza Waleed
PhD Fellow | GDE in Earth Engine
@waleedgeo.com





Mirza Waleed

PhD Researcher | Google Developer Expert

Department of Geography, Hong Kong
Baptist University



I am Waleed, a [PhD Fellow](#) in Geography focusing on remote sensing, geo-analytics, cloud computing, and artificial intelligence for disaster risk management, particularly for flood hazard. Besides, I am a [Google Developer Expert in Earth Engine](#) category. I am also working part-time on [Fiverr](#), where I provide geospatial analytics and Google Earth Engine related consultancy services.

My ultimate goal is to develop new (open-source) techniques or methods to improve the accuracy and effectiveness of flood hazard mapping, monitoring, and risk assessment. I am also interested in creating open-sourced tools, which can simplify existing geospatial analysis. I am always enthusiastic about connecting with like-minded professionals who share my passion for geospatial analytics, and I welcome any opportunity to collaborate and create a positive impact together.

Interests

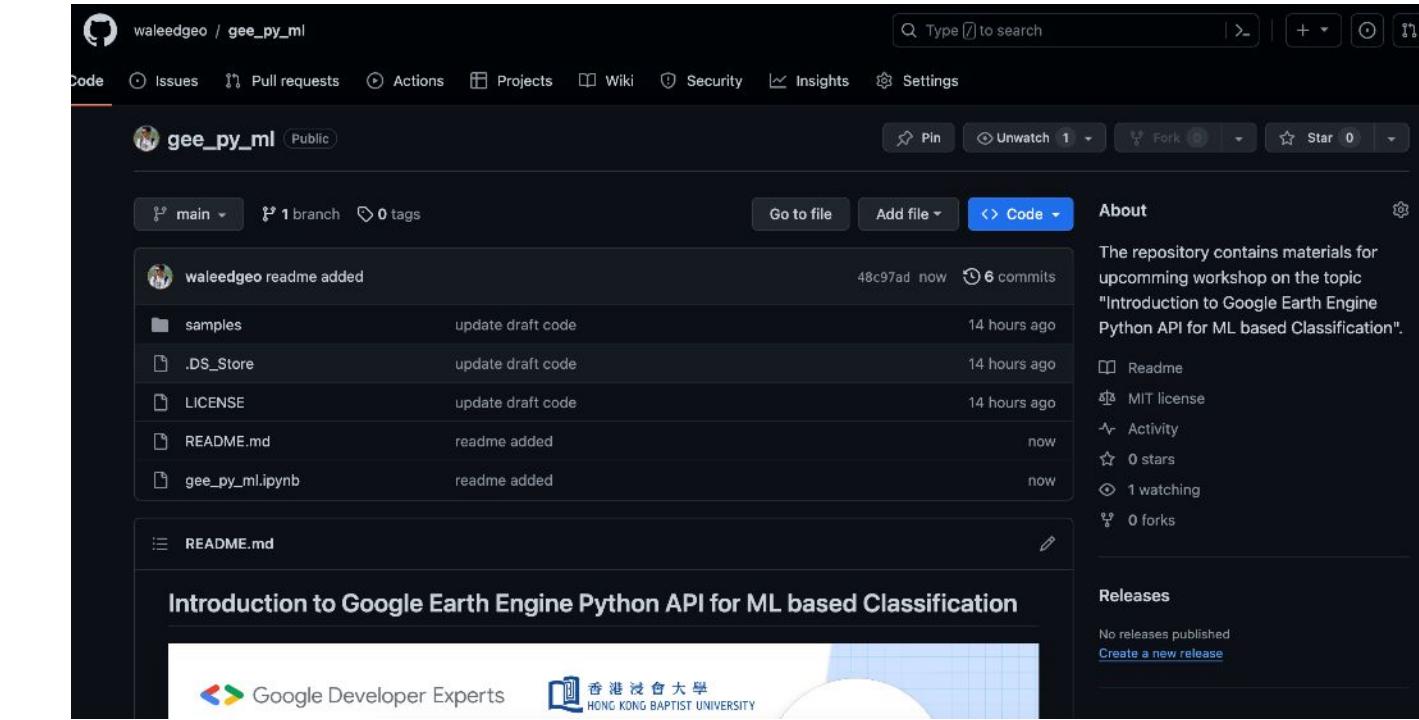
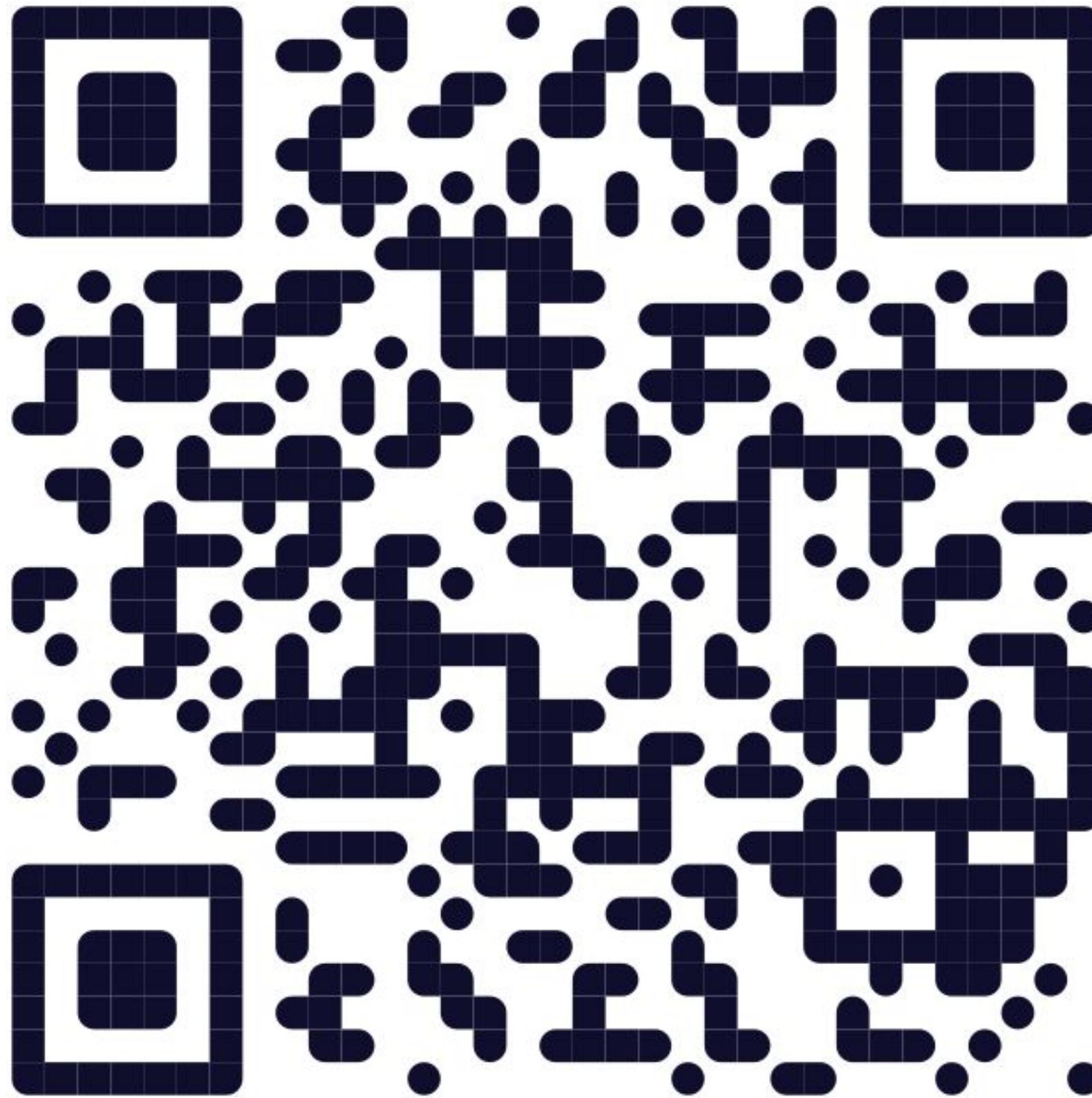
- Remote Sensing & GIS
- Applied Machine Learning
- Google Earth Engine
- Disaster Risk Management (Floods)

Education

- PhD in Geography (cont.), 2026
Hong Kong Baptist University - HK
- BS in Environmental Sciences, 2021
COMSATS University Islamabad - PK

waleedgeo.com

Project Files (Scan it)



This repository contains materials for the workshop on Google Earth Engine Python API for ML-based Classification. The session recording and presentation slides will be made available after the workshop (9-12-2023)

Click the link below to open the notebook in Google Colab

 Open in Colab

The notebook covers the following topics:

1. Why Python API for Google Earth Engine?
2. How to learn Earth Engine Python API?
3. Setting up your environment
4. Machine Learning Overview
5. Supervised Classification in Earth Engine

https://github.com/waleedgeo/gee_py_ml

Agenda

01 GEE Python API?

Why and Where to learn?

02 Pre-requirements

Environment Management & IDEs

03 Machine Learning Overview

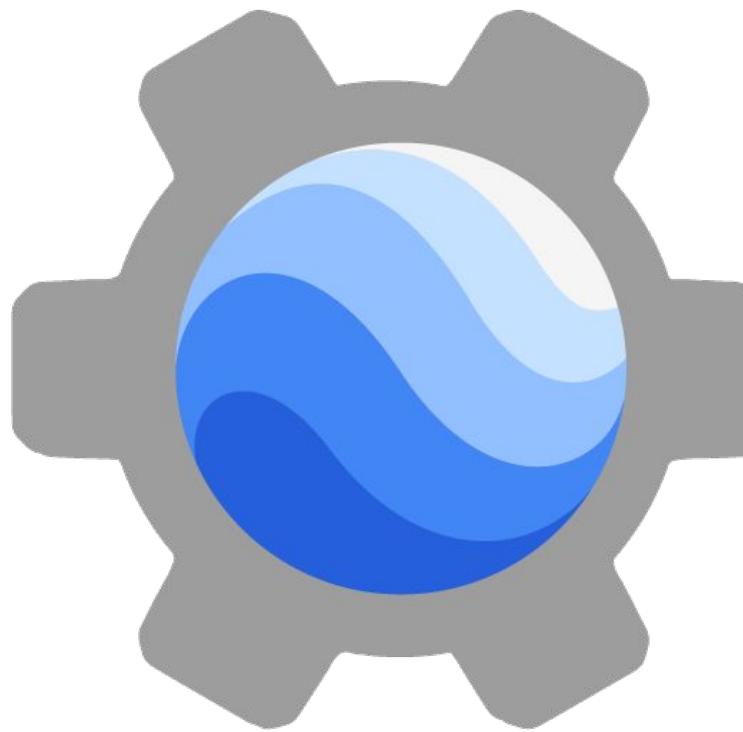
Overview, and Models Types

04 Code Session (Workflow-1)

Server-side ML-based classification in
GEE using Geemap

05 Q&A

GEE Python API



Google Earth Engine



Databases

+



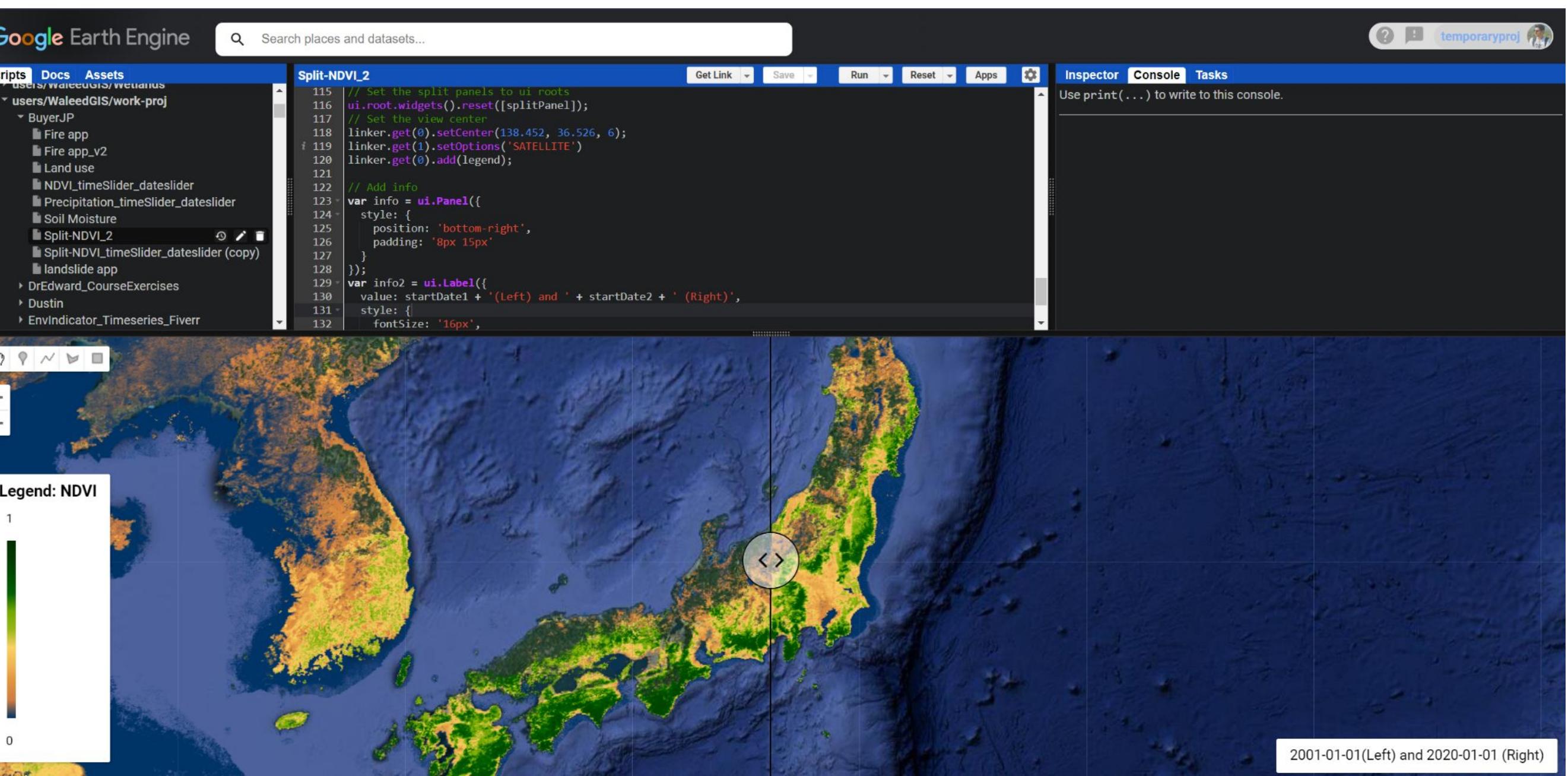
Processing

+



API

JavaScript API



Python API

The screenshot shows a Jupyter Notebook cell with the title '09_timelapse.ipynb'. The code cell contains Python code using the Geemap library to perform various operations. The code includes importing ee and geemap, initializing the Earth Engine API, creating a sequence of numbers, defining a function to compute squares, applying the map function to a list, and creating cloud-free composites. The code is displayed in a dark-themed notebook interface.

```
# %pip install pygis
import ee
import geemap

geemap.ee_initialize()

# The map function
myList = ee.List.sequence(1, 10)
myList

def computeSquares(number):
    return ee.Number(number).pow(2)

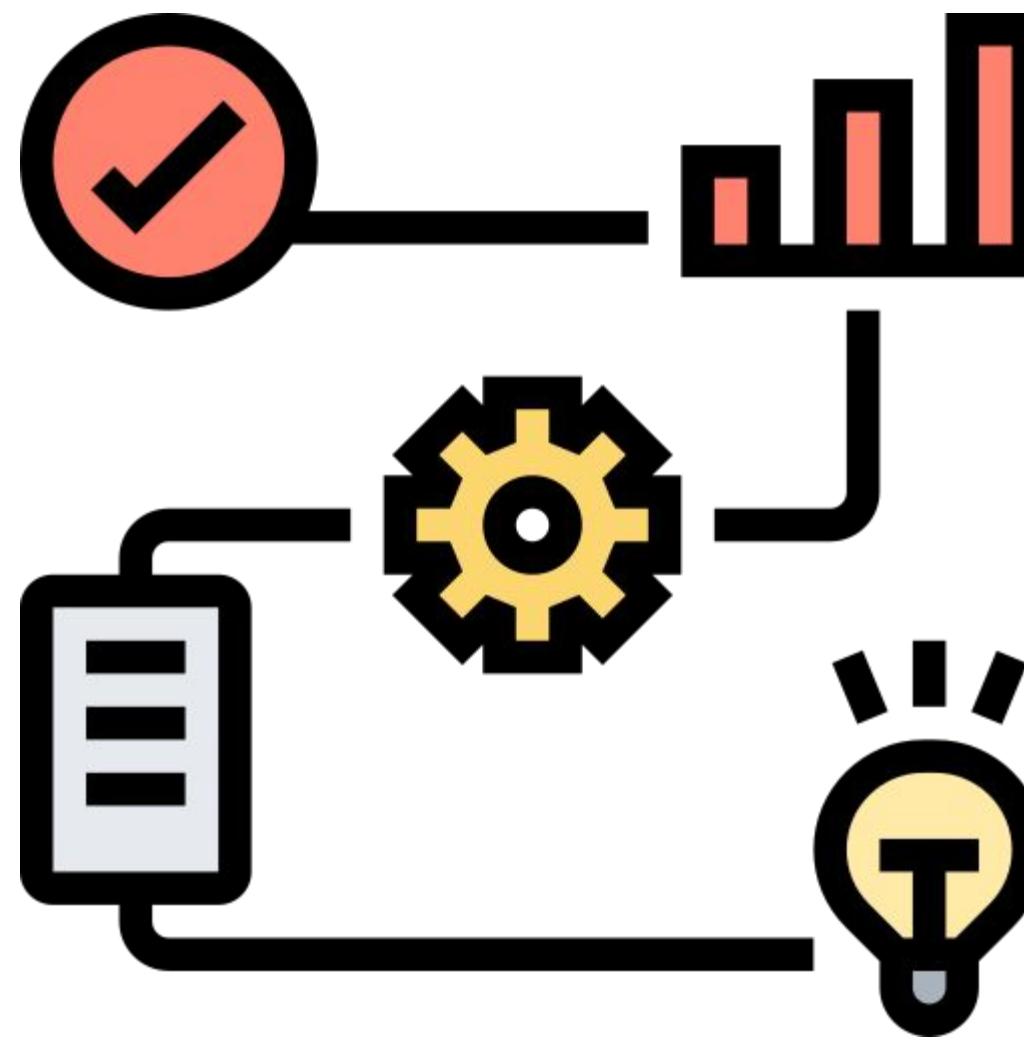
squares = myList.map(computeSquares)
squares

# squares = myList.map(lambda number: ee.Number(number).pow(2))
squares

# Creating cloud-free composites
Map = geemap.Map()
fc = ee.FeatureCollection('USDOOS/LSIB_SIMPLE/2017').filter(
    ee.Filter.eq('country_na', 'Netherlands')
)
Map.addLayer(fc, {'color': 'ff000000'}, "Netherlands")
Map.centerObject(fc)
Map

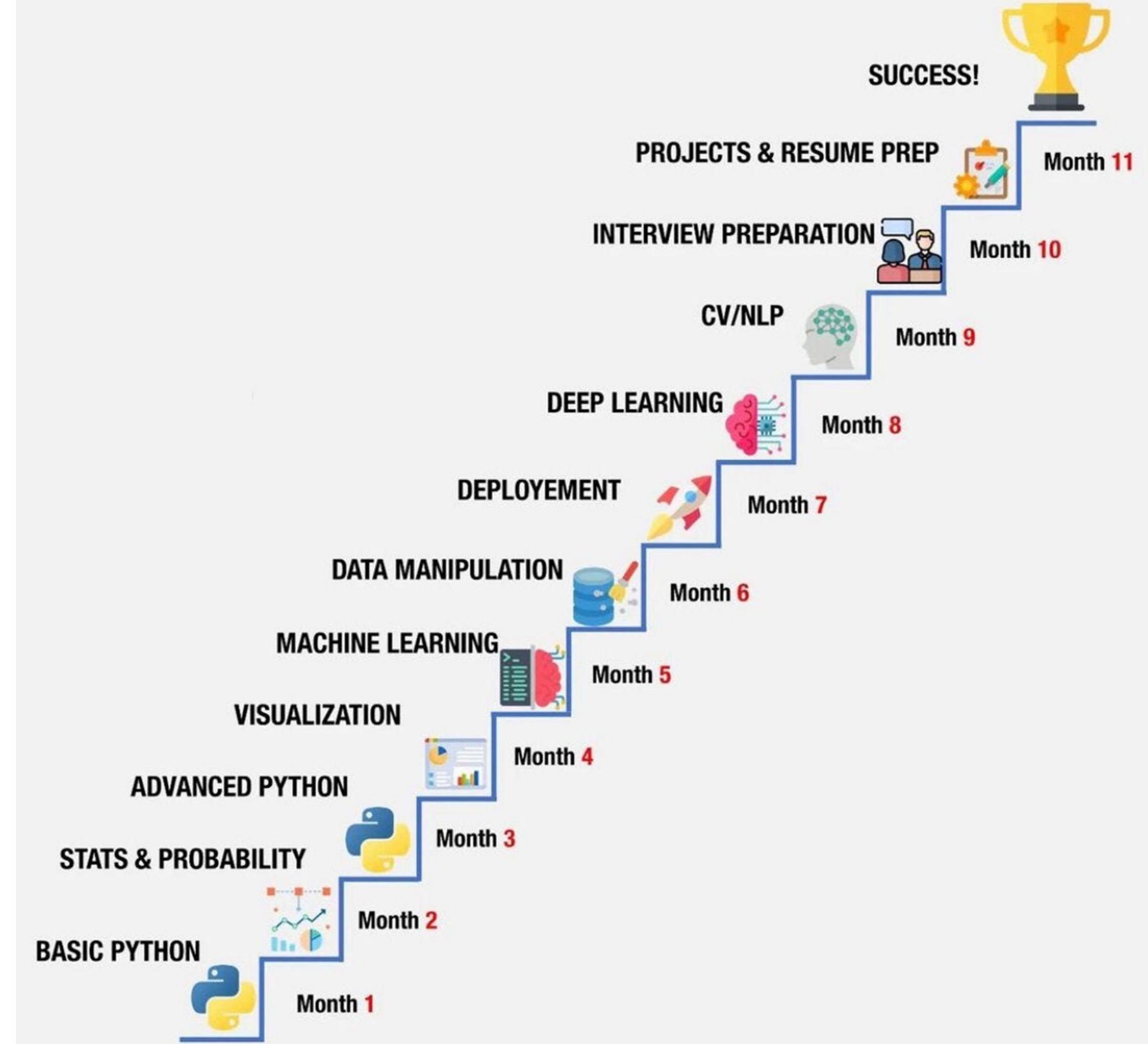
years = ee.List.sequence(2013, 2022)
```

Why Python?

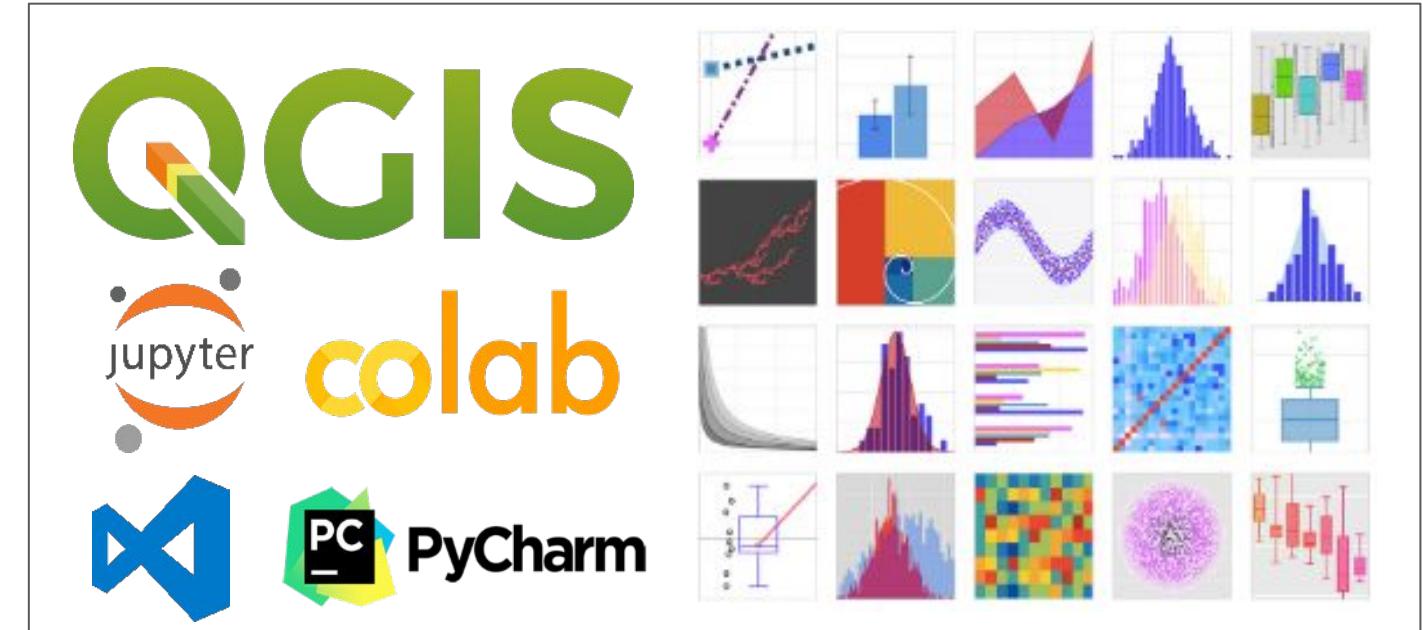


Geospatial Data Science

DATA SCIENCE ROADMAP



Why choose Python + Earth Engine?



01

Automation

Run scripts without user intervention.

02

Libraries

Tensorflow
PyTorch
scikit-learn
Data Analysis
Data Visualization
Physics Libraries

03

Cloud Services

BigQuery
Dataflow / Beam
Vertex AI
Google Cloud Storage

04

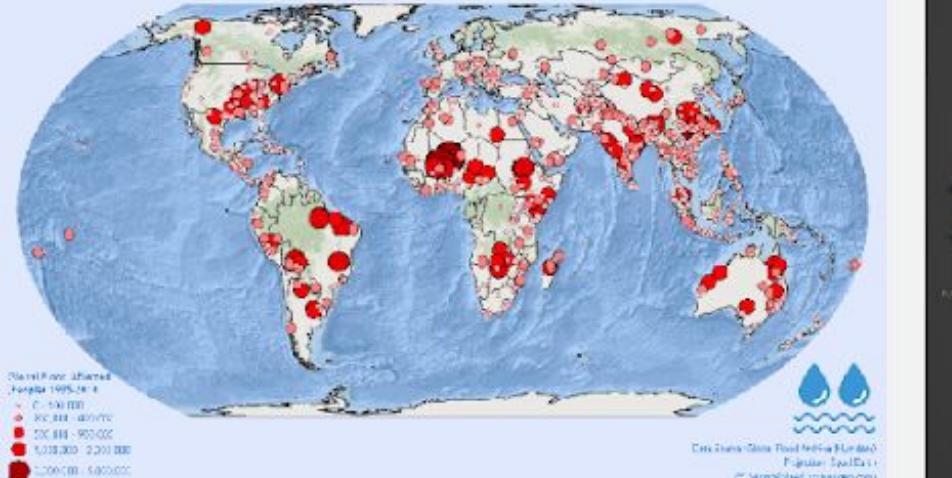
Dev Tools

GitHub
CI/CD Tools
Testing Frameworks

Source:
GeoForGood23

GLOBAL FLOOD

AFFECTED PEOPLE



DENMARK

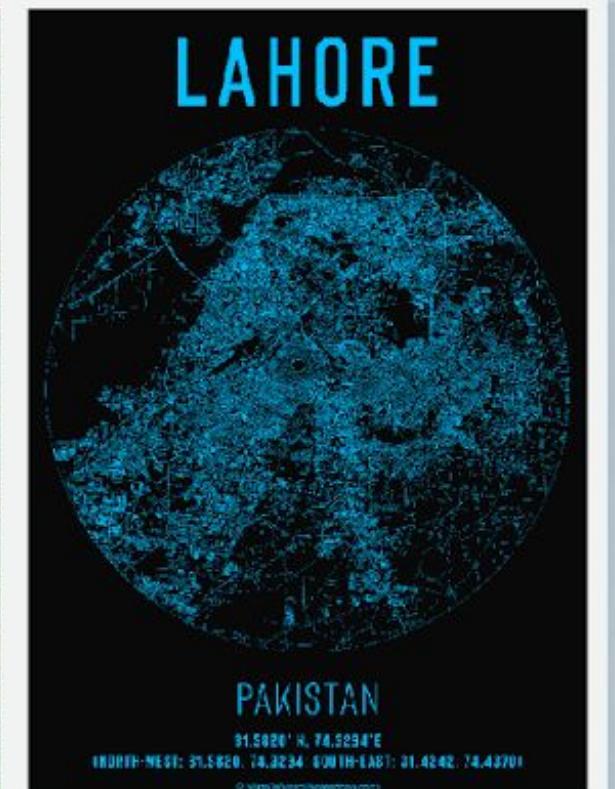
Population: 5.6 Million



YOSEMITE
Wilderness

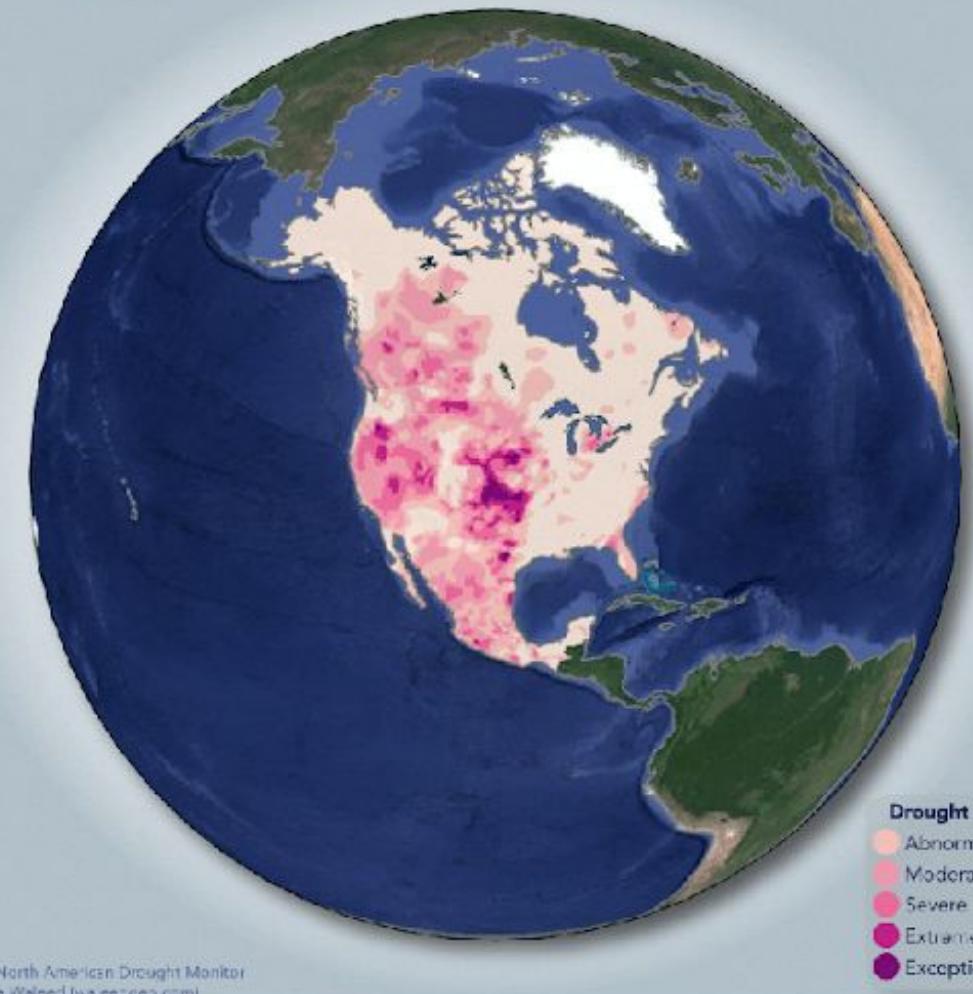


LAHORE



DROUGHT IN NORTH AMERICA 2023

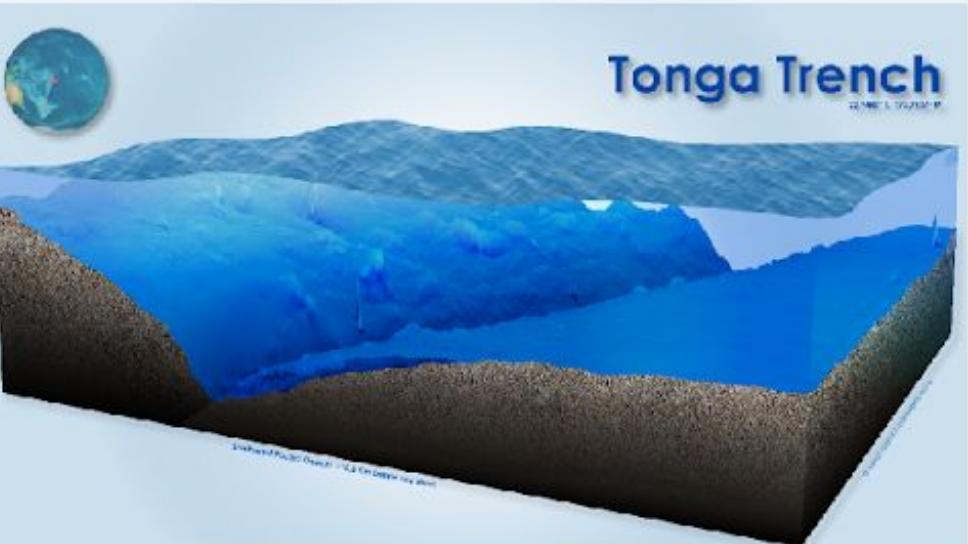
JANUARY FEBRUARY MARCH APRIL MAY JUNE JULY AUGUST SEPTEMBER



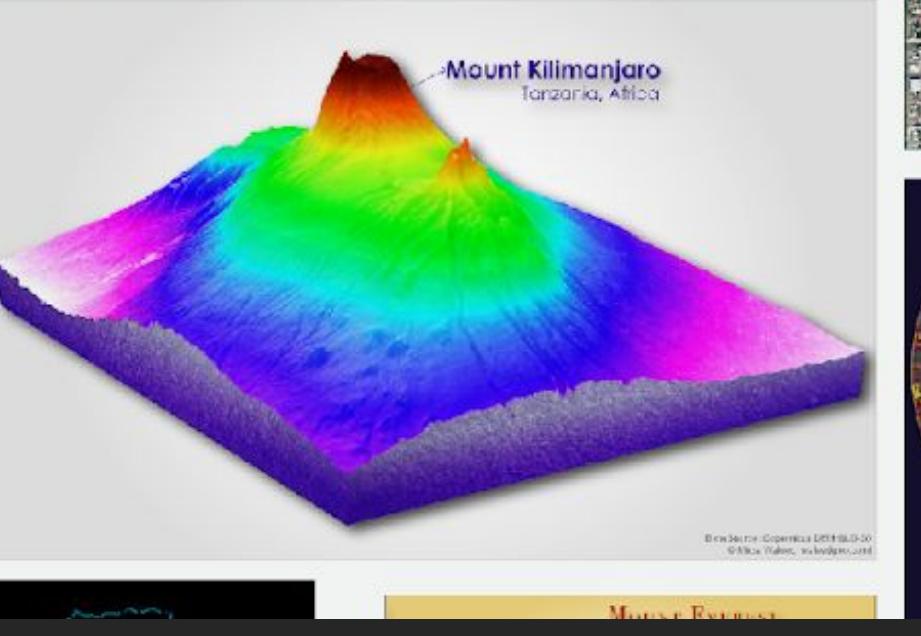
MIGRATION FLOW



Tonga Trench



Mount Kilimanjaro



MANAUS

Where Civilization Meets Wilderness

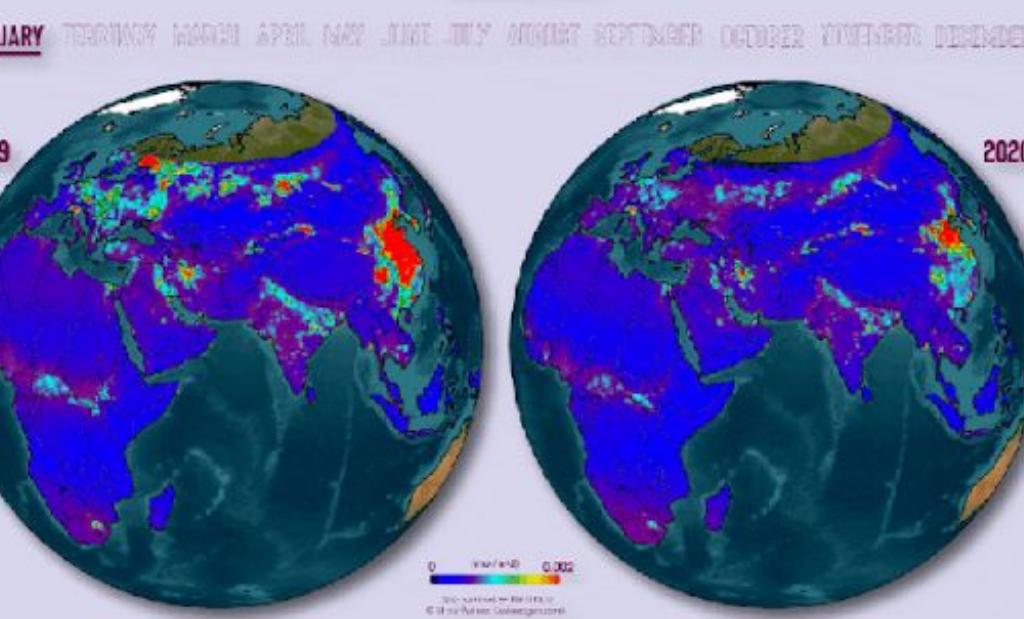
NORTHWEST: 51.5000° N, 70.4250° E | SOUTHEAST: 31.4242° N, 74.4370° E

© Mirza Waheed (waheedgeo.com)

PAKISTAN

© Mirza Waheed (waheedgeo.com)

Nitrogen Dioxide (NO₂) Concentration



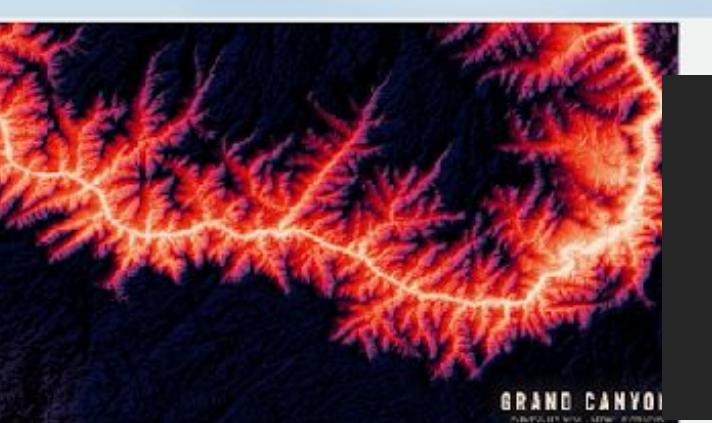
“EE Python API Example”

My Journey of 30-Day Map Challenge (2023)



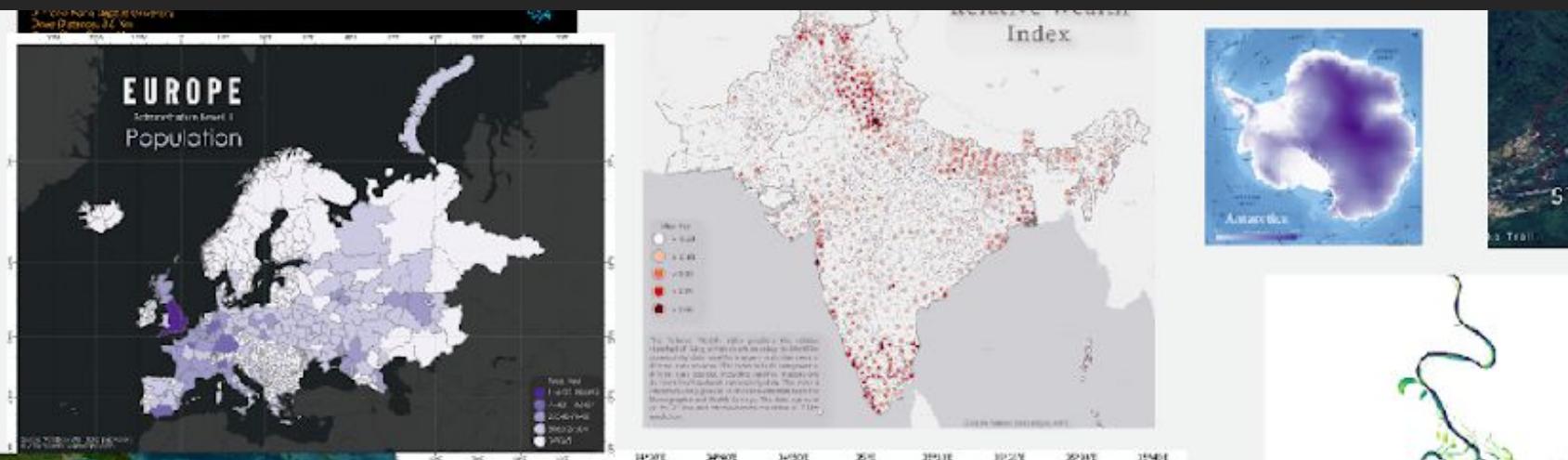
ASIA

RIVER INDEX



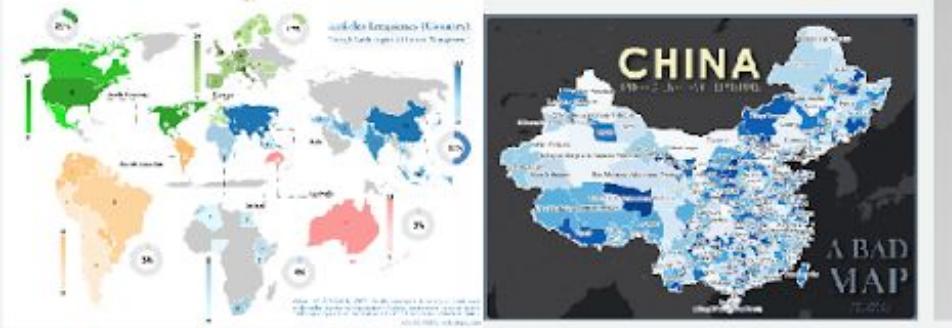
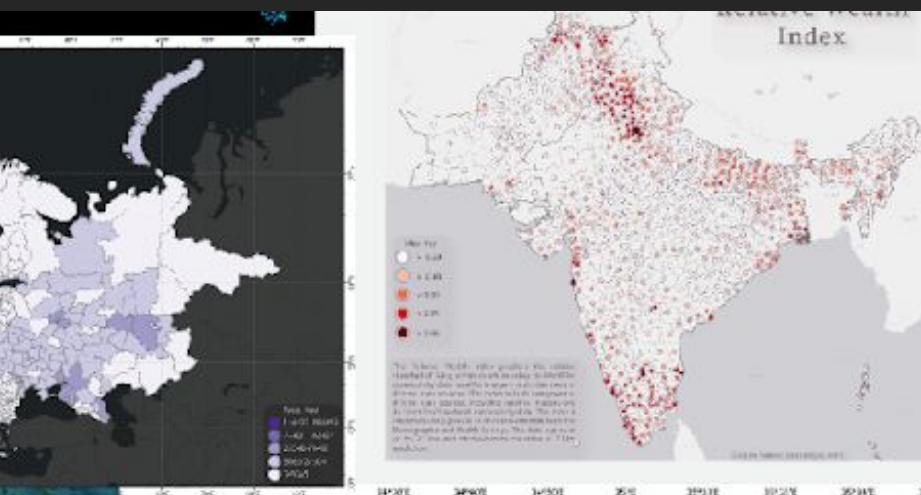
GRAND CANYON

Picture



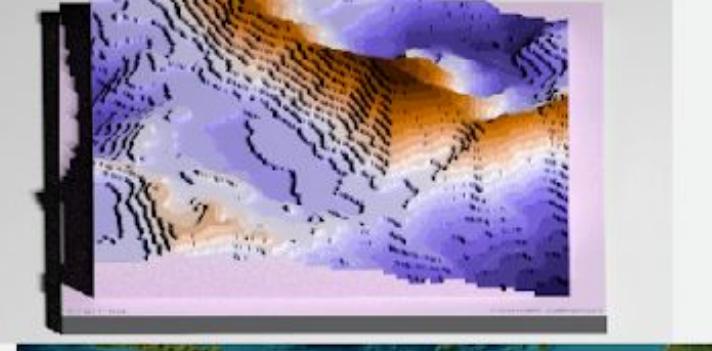
EUROPE

Population



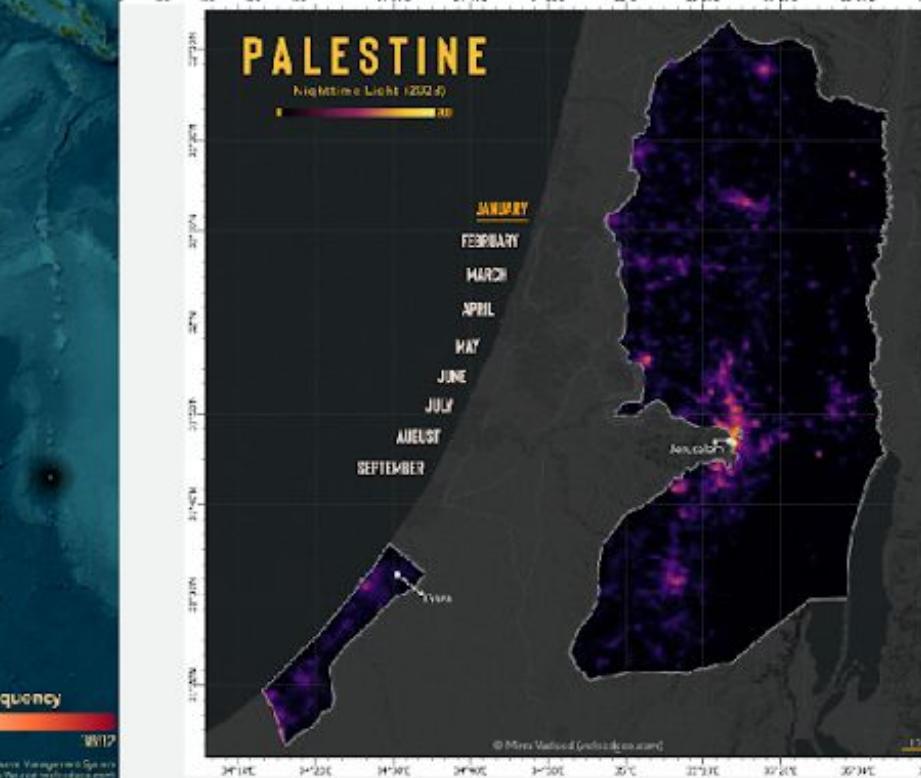
CHINA

ABAD MAP



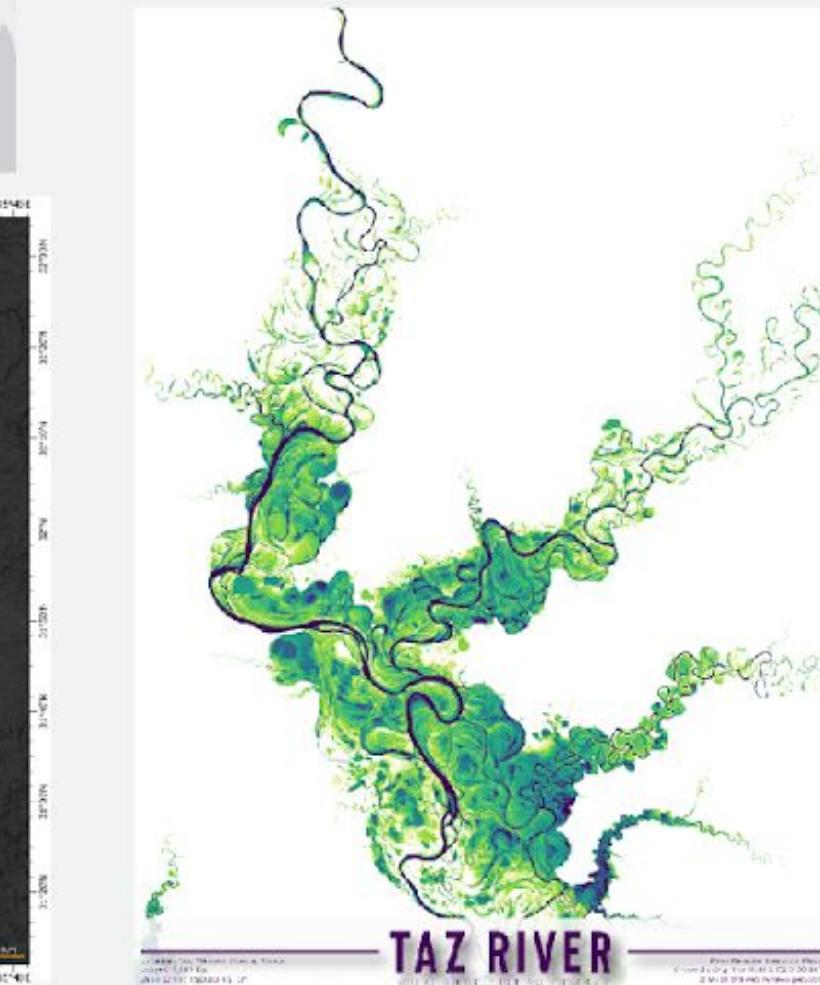
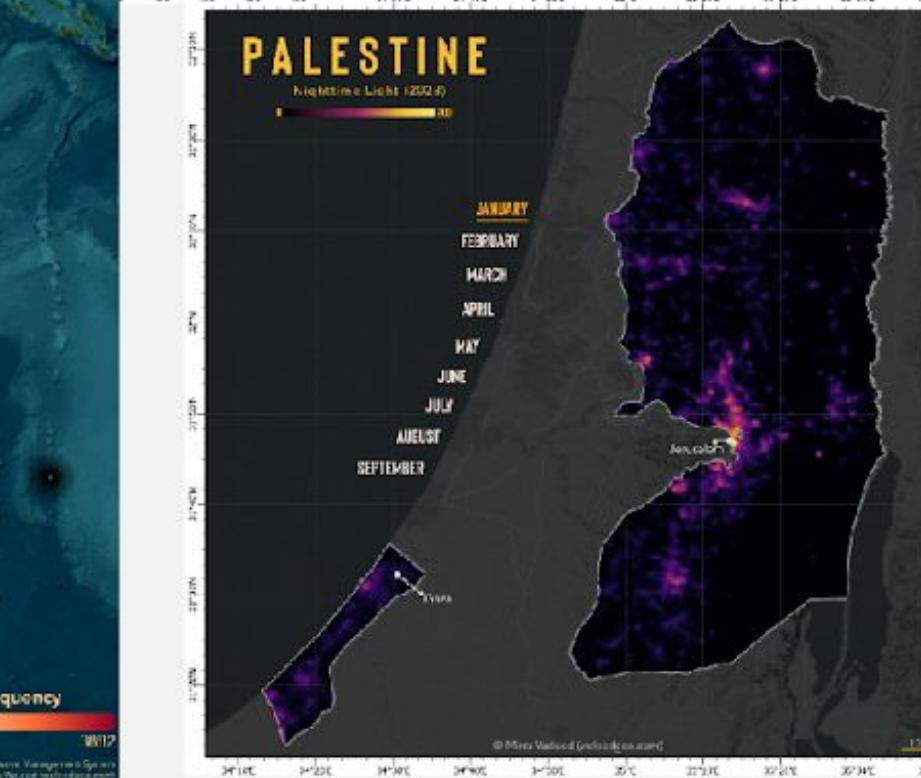
NGARI

Picture

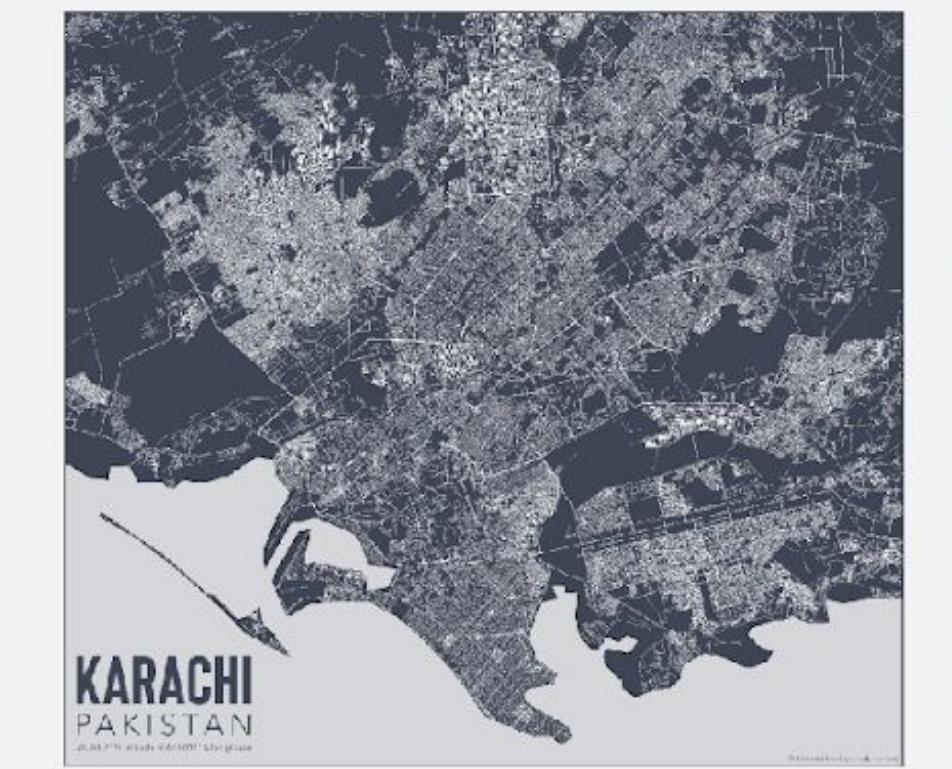
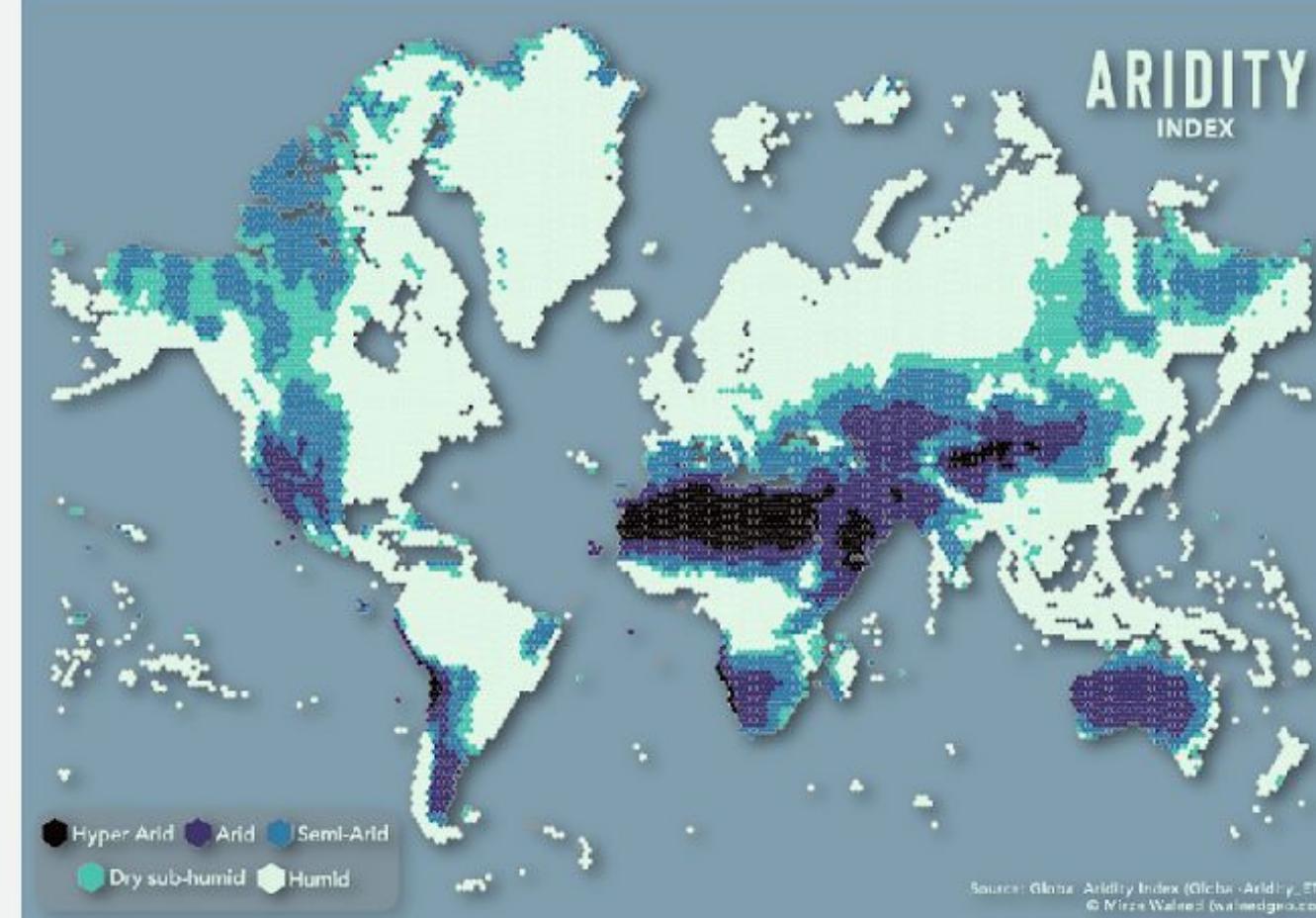


PALESTINE

Nighttime Light (2000-10)



TAZ RIVER



KARACHI

PAKISTAN



AUSTRALIA

Forest Fire Frequency

2001-2023

Frequency

WRI

© Mirza Waheed (waheedgeo.com)

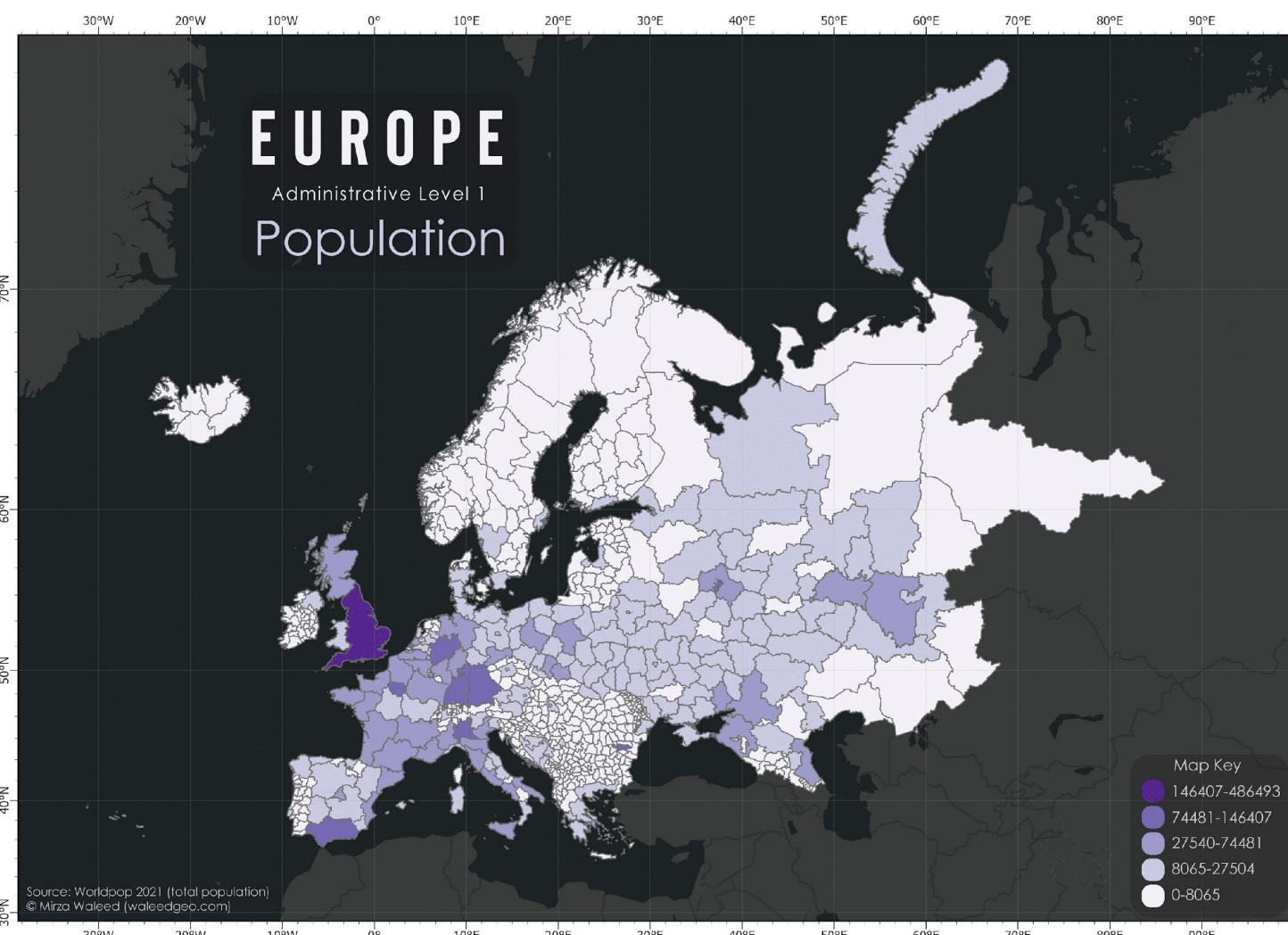
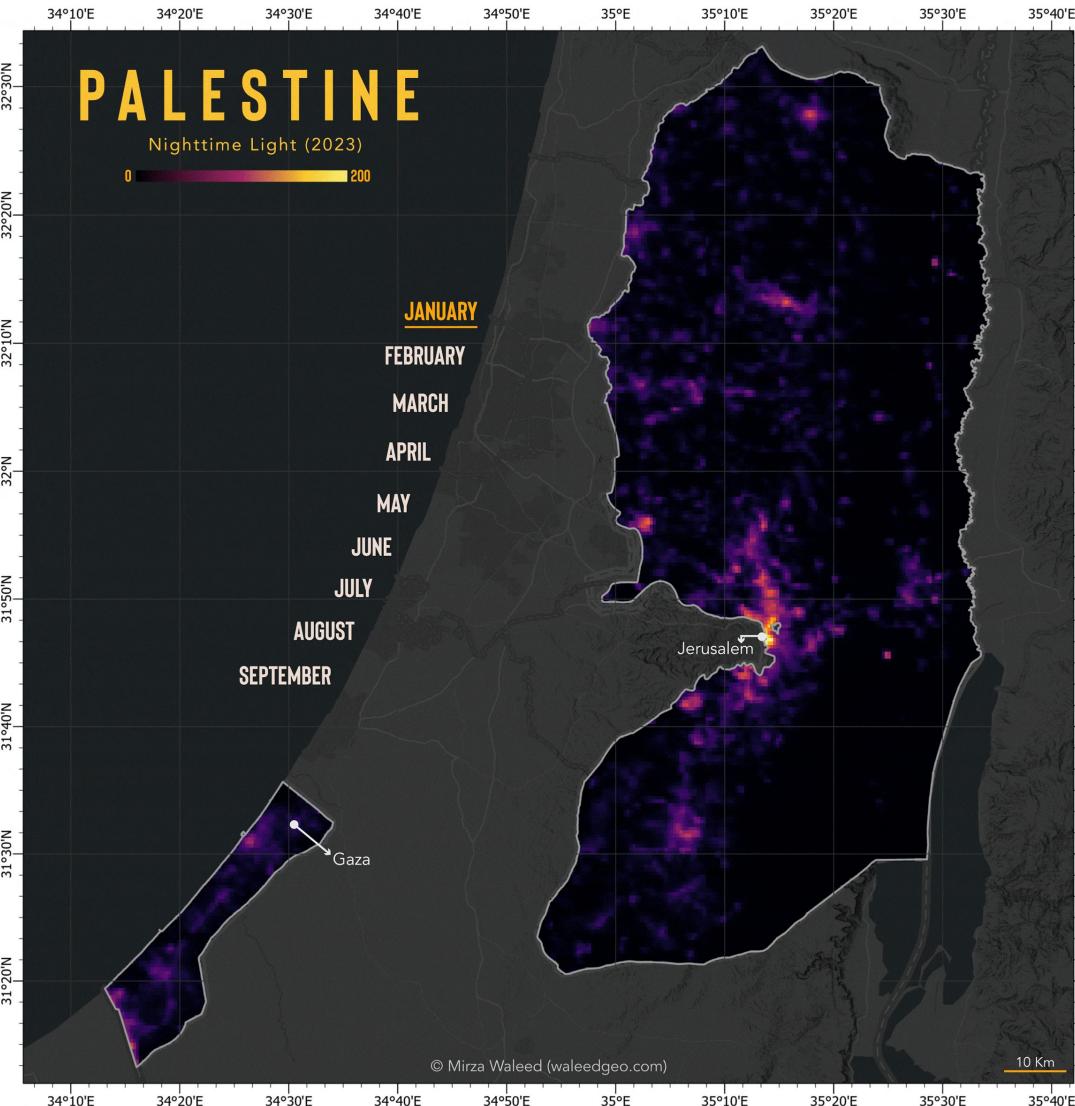
ARIDITY

INDEX

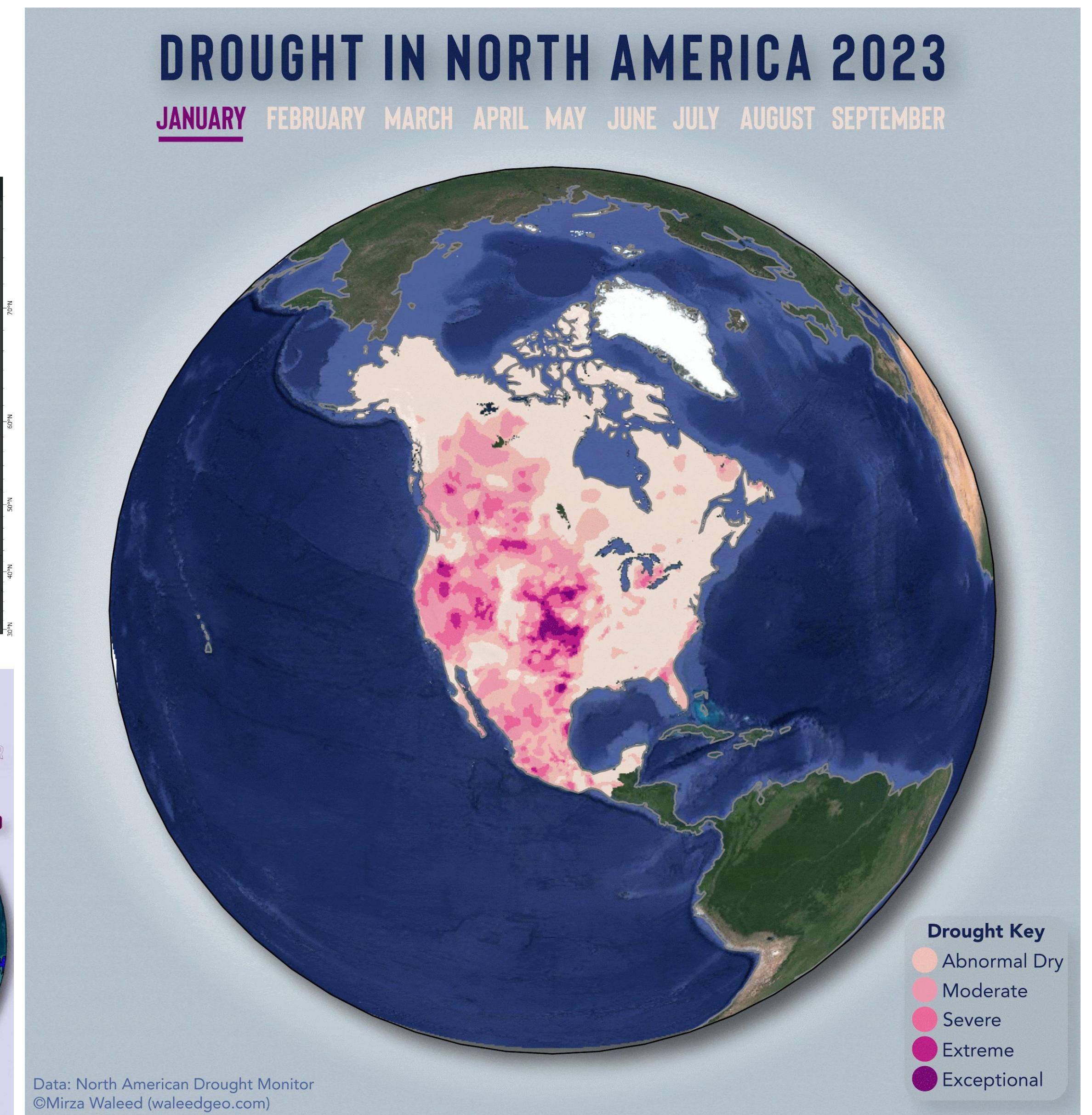
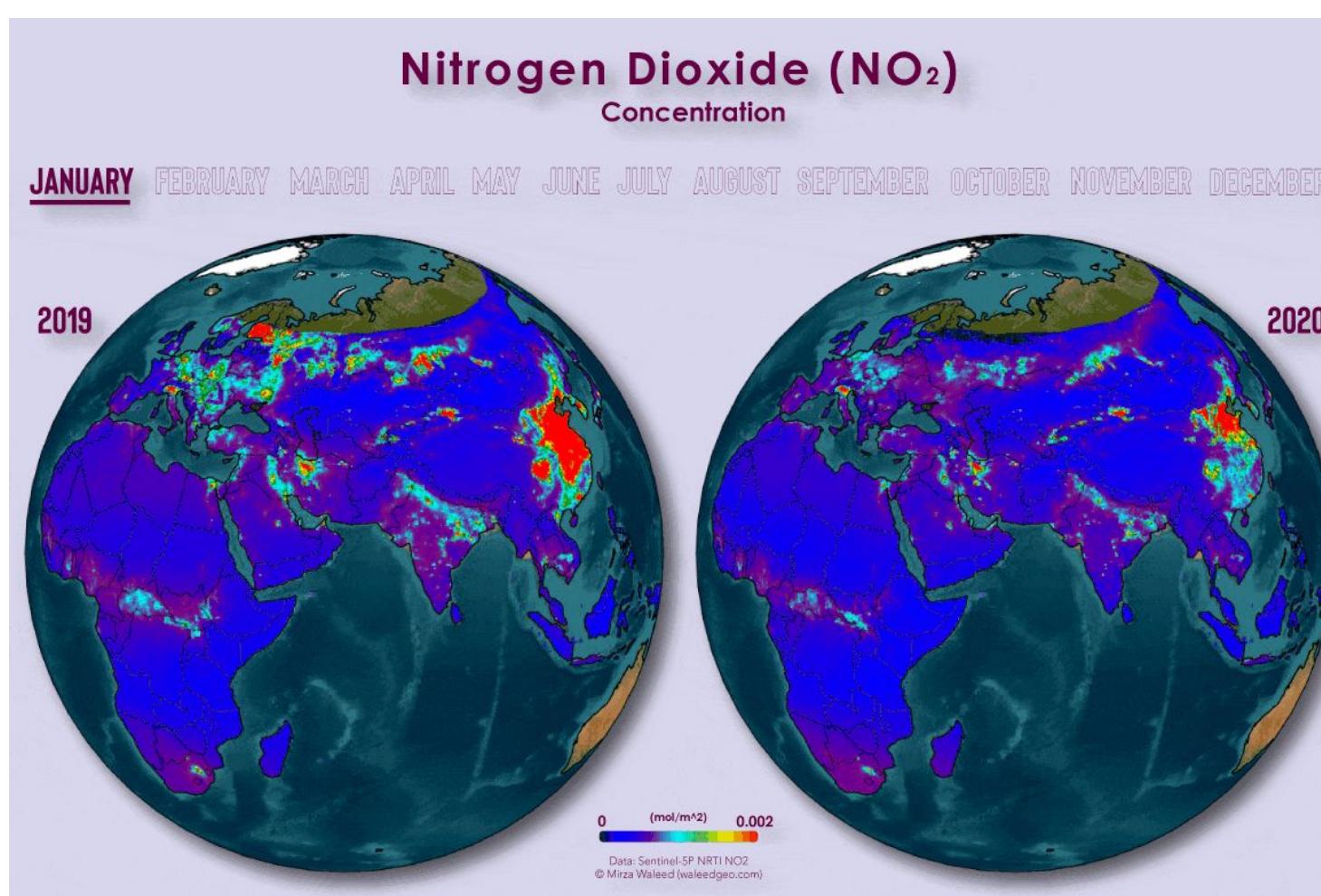
Source: Gleason-Aridity Index (Gleason-Aridity, ETO)

© Mirza Waheed (waheedgeo.com)

Made with Geemap & Cartoee



My 30 Day Map Challenge (2023)



Earth Engine JavaScript and Python APIs use nearly identical syntax to describe data classes and their methods.

Classes

`ee.Image`

Methods

`.clip(..)`
`.sample(..)`
`.reduceRegion(..)`

`ee.ImageCollection`

`.filter(..)`
`.reduce(..)`
`.toBands(..)`

`ee.Feature`

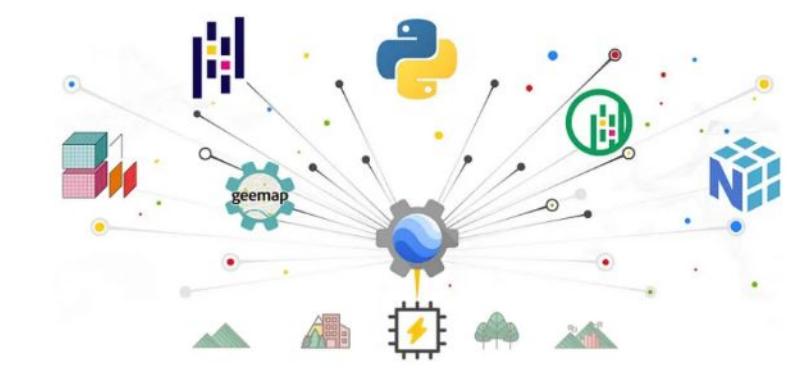
`.buffer(..)`
`.simplify(..)`
`.area(..)`



Where to Learn GEE Python API?

Resources

- [Python Powers Up: The Rise of the Python API for Earth Engine](#)
- [Beginners crash course of Python in Earth Engine for Environmental Insights |Geo for Good 2023](#)
- [Geemap](#)
- [An Intro to the Earth Engine Python API](#)



The Earth Engine Python experience powers up, expanding its connectivity within the Python science ecosystem. Source: Google

[Python Powers Up: The Rise of the Python API for Earth Engine](#)

 Google Earth · Follow
Published in Google Earth and Earth Engine · 5 min read · Oct 26

Welcome to geemap

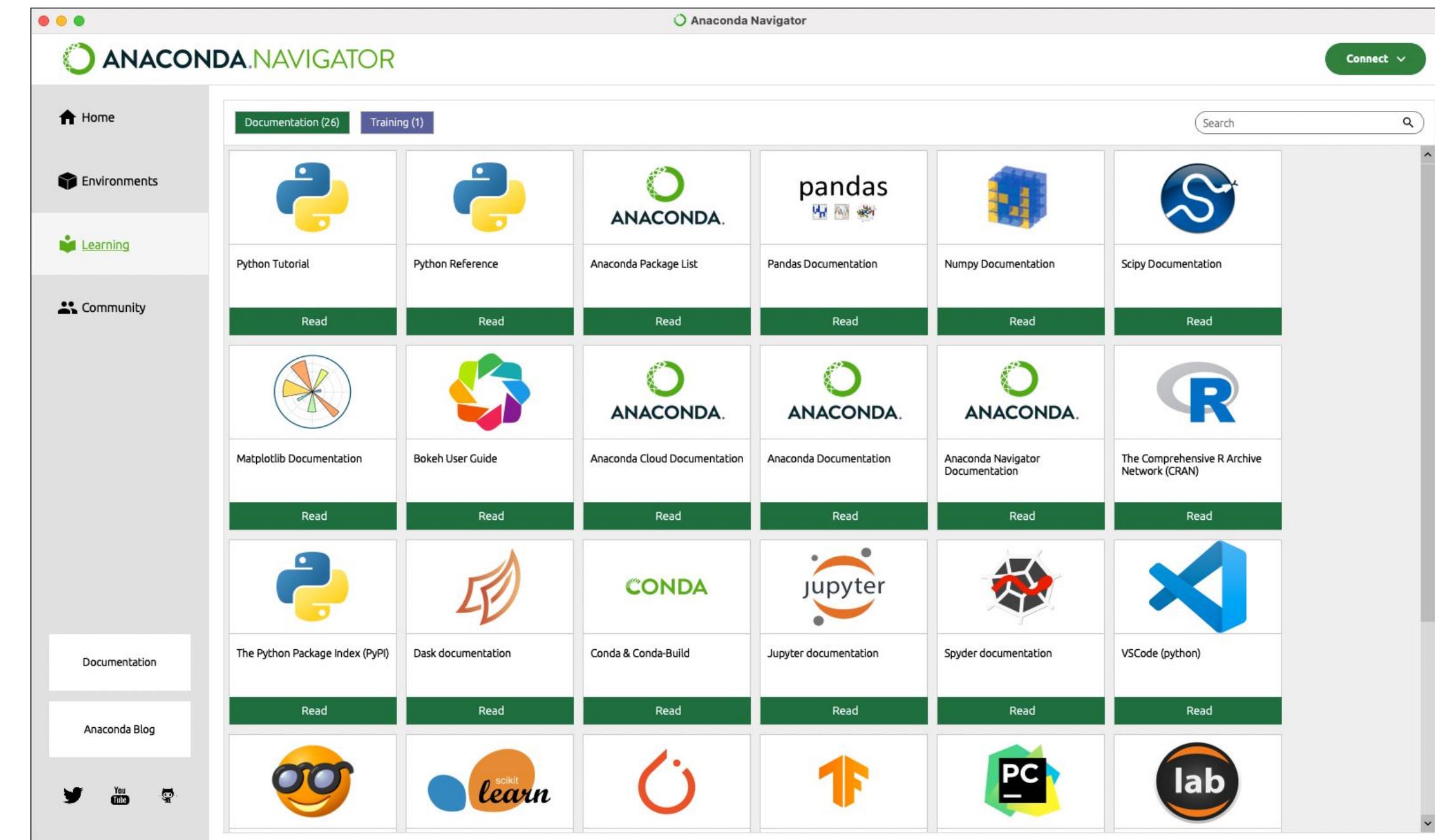
A Python package for interactive geospatial analysis and visualization with Google Earth Engine

- GitHub repo: <https://github.com/gee-community/geemap>
- Documentation: <https://geemap.org>
- PyPI: <https://pypi.org/project/geemap>
- Conda-forge: <https://anaconda.org/conda-forge/geemap>
- 350+ GEE notebook examples: <https://github.com/giswqs/erthengine-py-notebooks>
- GEE Tutorials on YouTube: <https://youtube.com/@giswqs>
- Free software: MIT license

Pre-requirements

Package Management: Conda

It is only required if you want to work locally instead of Google Colab.



[Tutorial: How to install Miniconda and geemap](#)

IDE: Google Colab

The image shows the Google Colab interface. At the top, there's a large "colab" logo. Below it, the main area has a "Welcome To Colaboratory" header with a "Sign in" button. The main content area contains a section titled "What is Colaboratory?" which explains that Colab allows writing and executing Python in a browser. It lists benefits like zero configuration, free access to GPUs, and easy sharing. Below this, there's a "Getting started" section with a code cell example:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

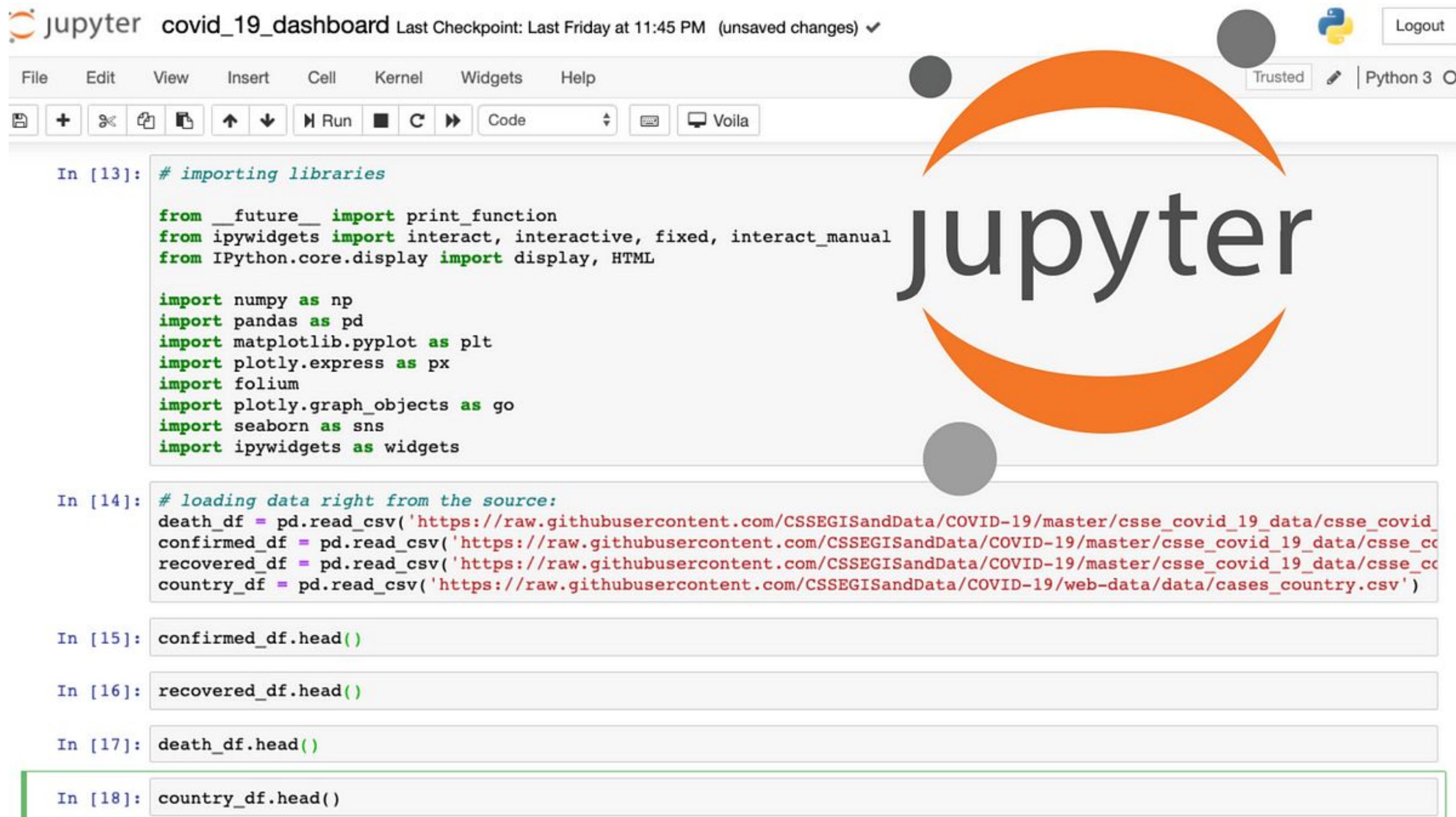
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

The image shows the Visual Studio Code interface. The title bar says "serviceWorker.js — create-react-app". The left sidebar shows a file tree for a "CREATE-REACT-APP" project, including files like App.css, App.js, App.test.js, index.css, index.js, logo.svg, serviceWorker.js, .gitignore, package-lock.json, package.json, README.md, and yarn.lock. The main editor pane shows the "serviceWorker.js" file with some code. The status bar at the bottom indicates "Compiled successfully!" and provides local and network URLs for the application. The bottom right corner shows the status bar with "Ln 34, Col 13" and other file details.

[Google Colab](#)

IDE: Jupyter Notebook



The image shows a Jupyter Notebook interface with a large orange smiley face logo overlaid on the right side. The notebook has several code cells:

- In [13]:

```
# importing libraries
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets
```
- In [14]:

```
# loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cc
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cc
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')
```
- In [15]:

```
confirmed_df.head()
```
- In [16]:

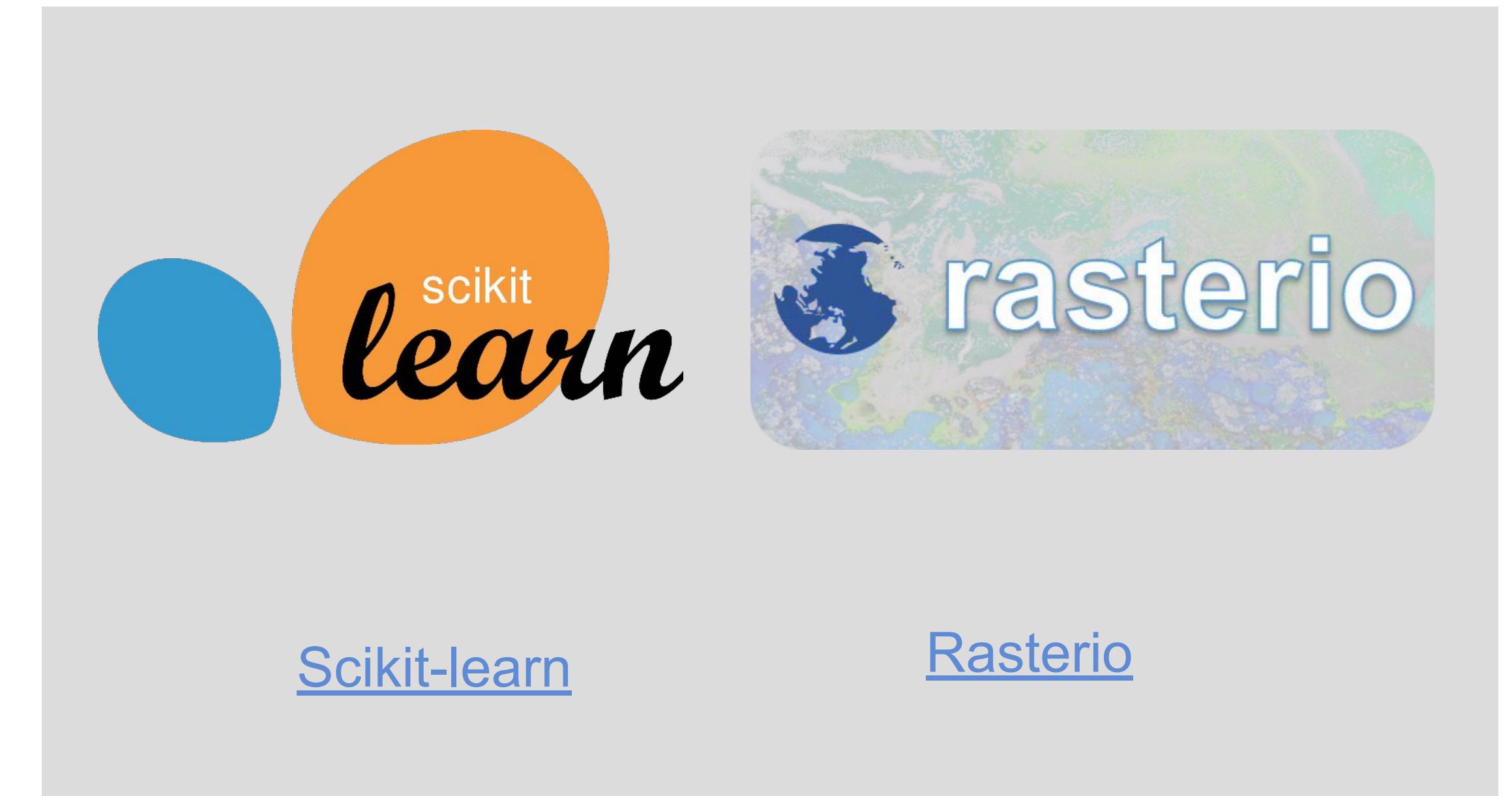
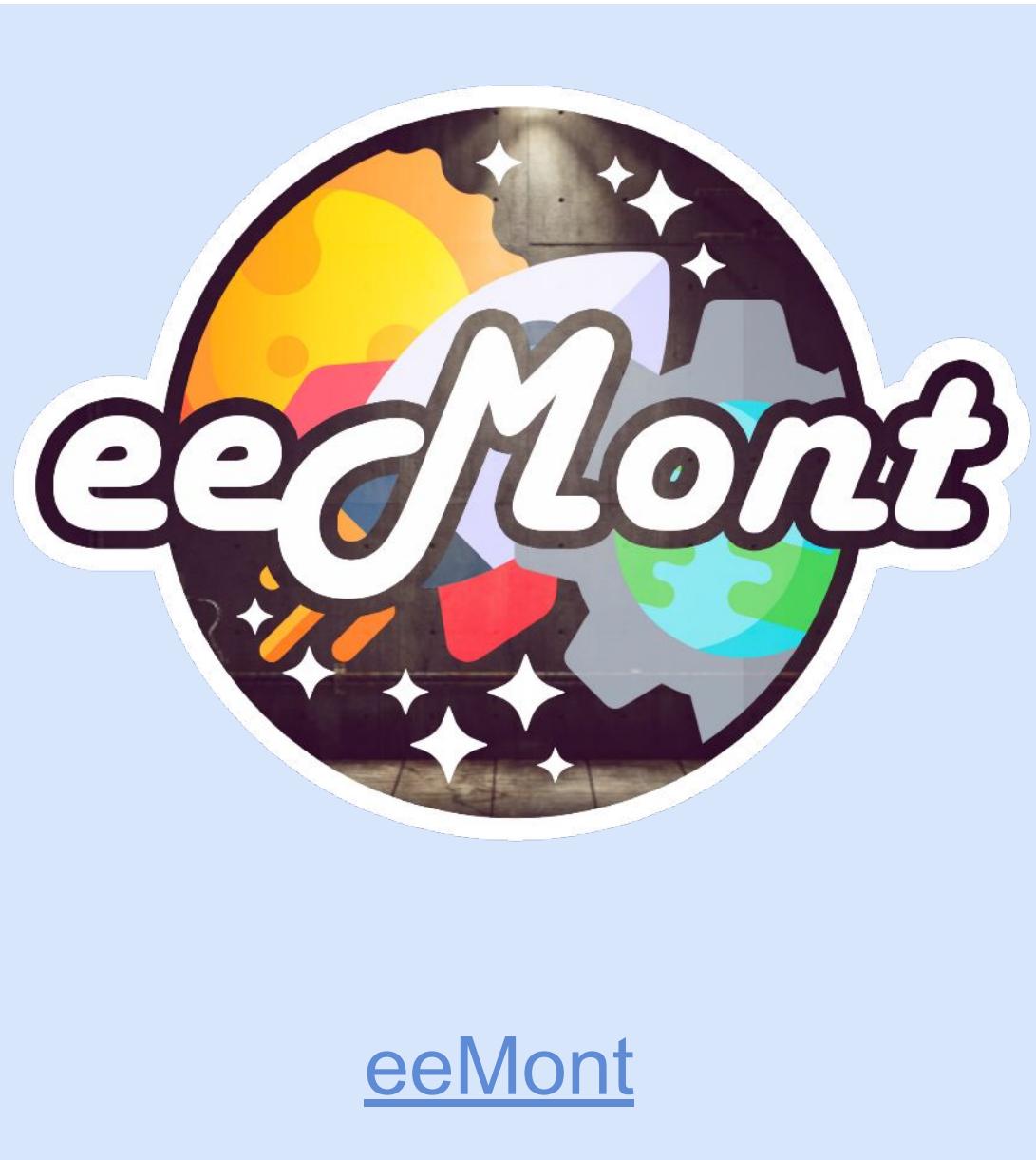
```
recovered_df.head()
```
- In [17]:

```
death_df.head()
```
- In [18]:

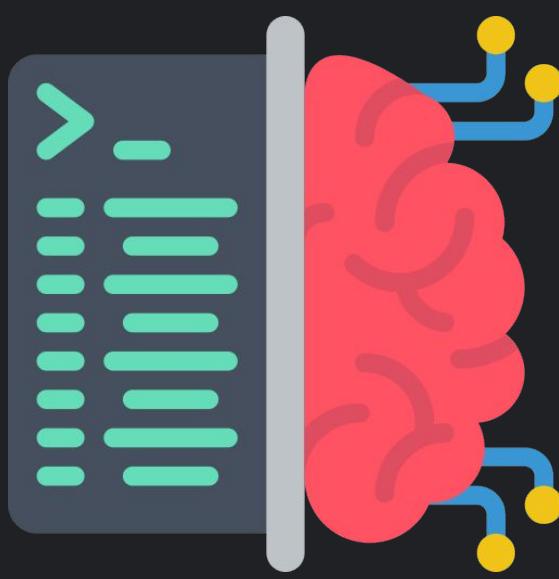
```
country_df.head()
```

[Download VSCode](#)

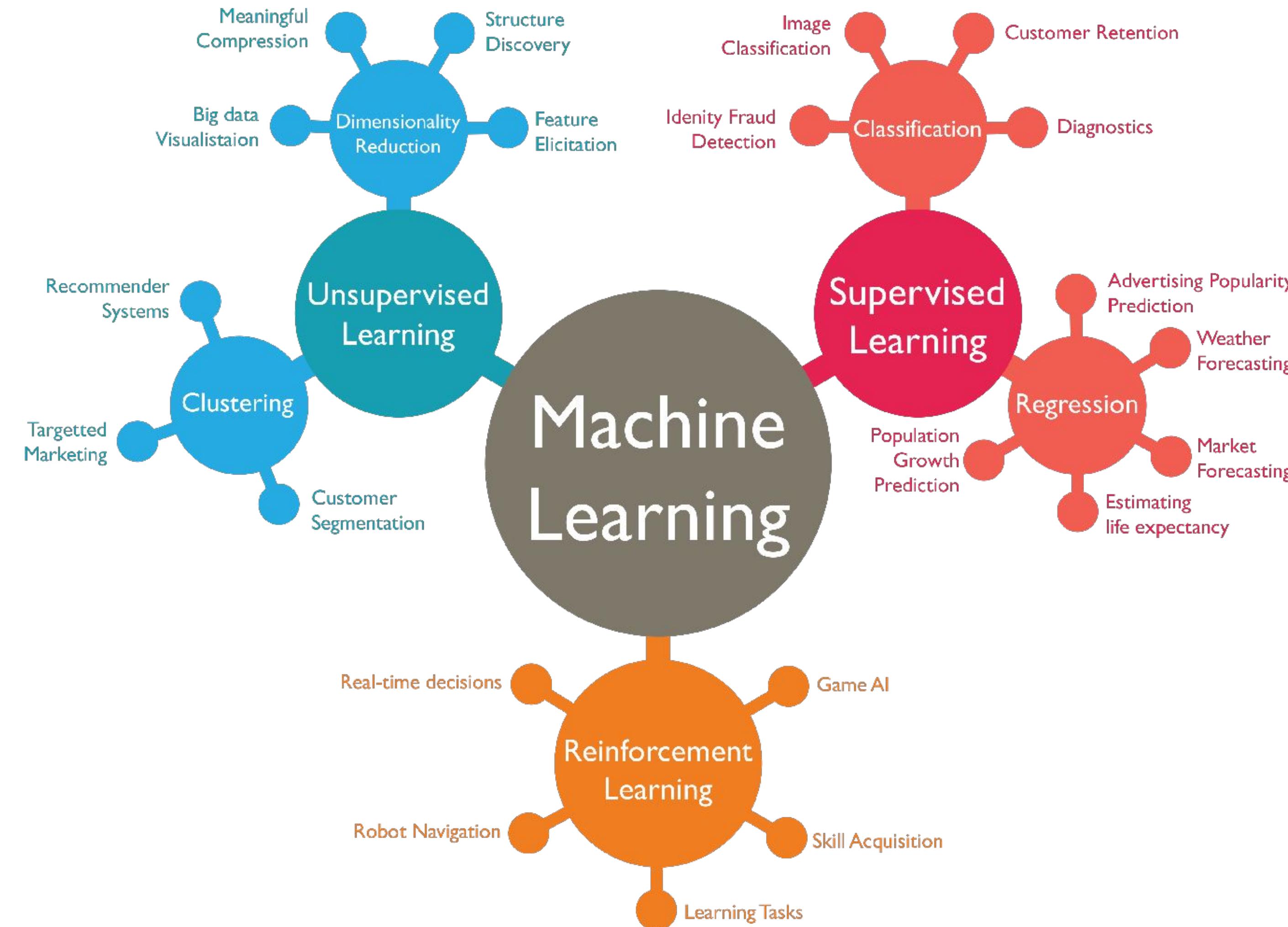
What Python packages will we use?



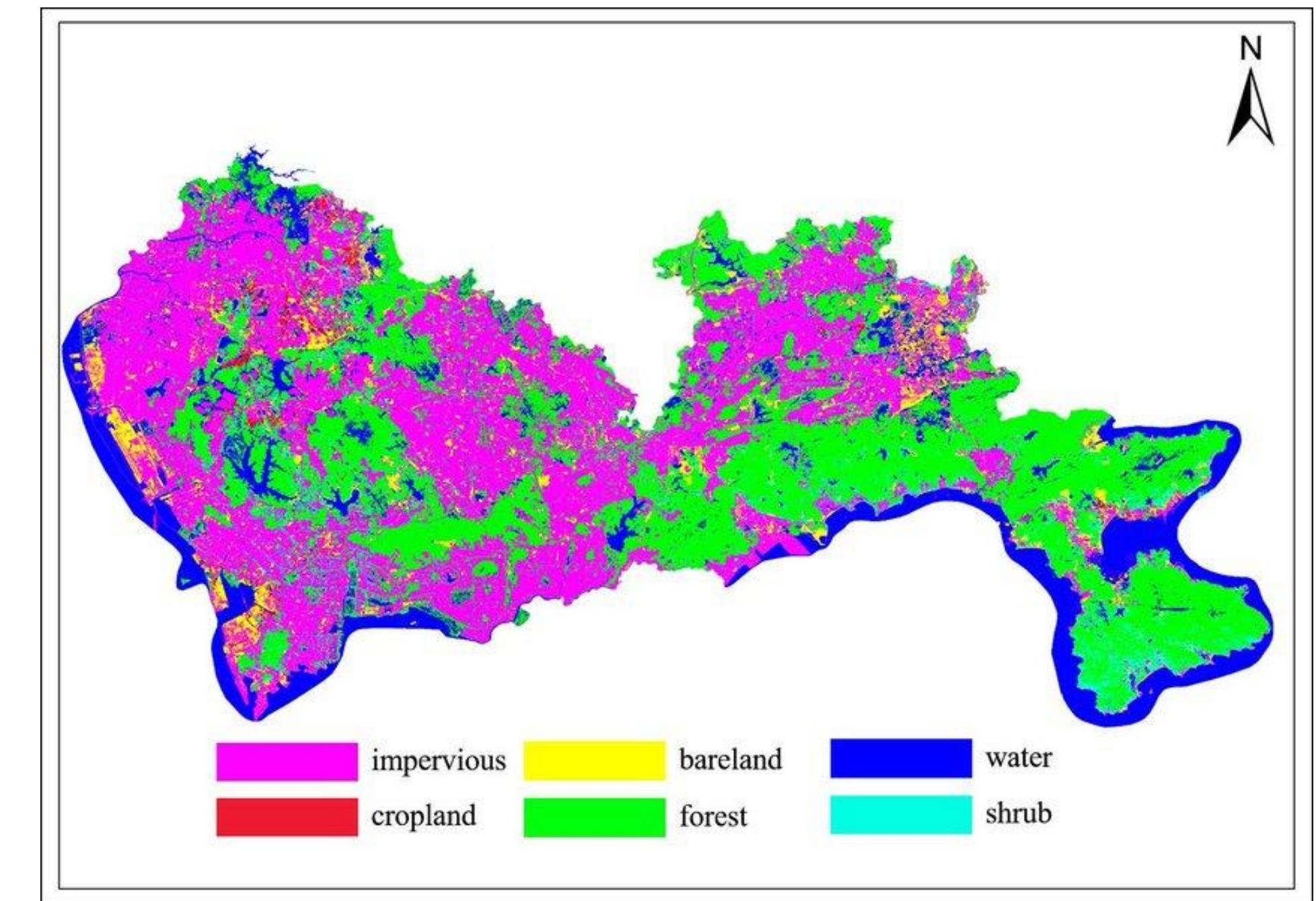
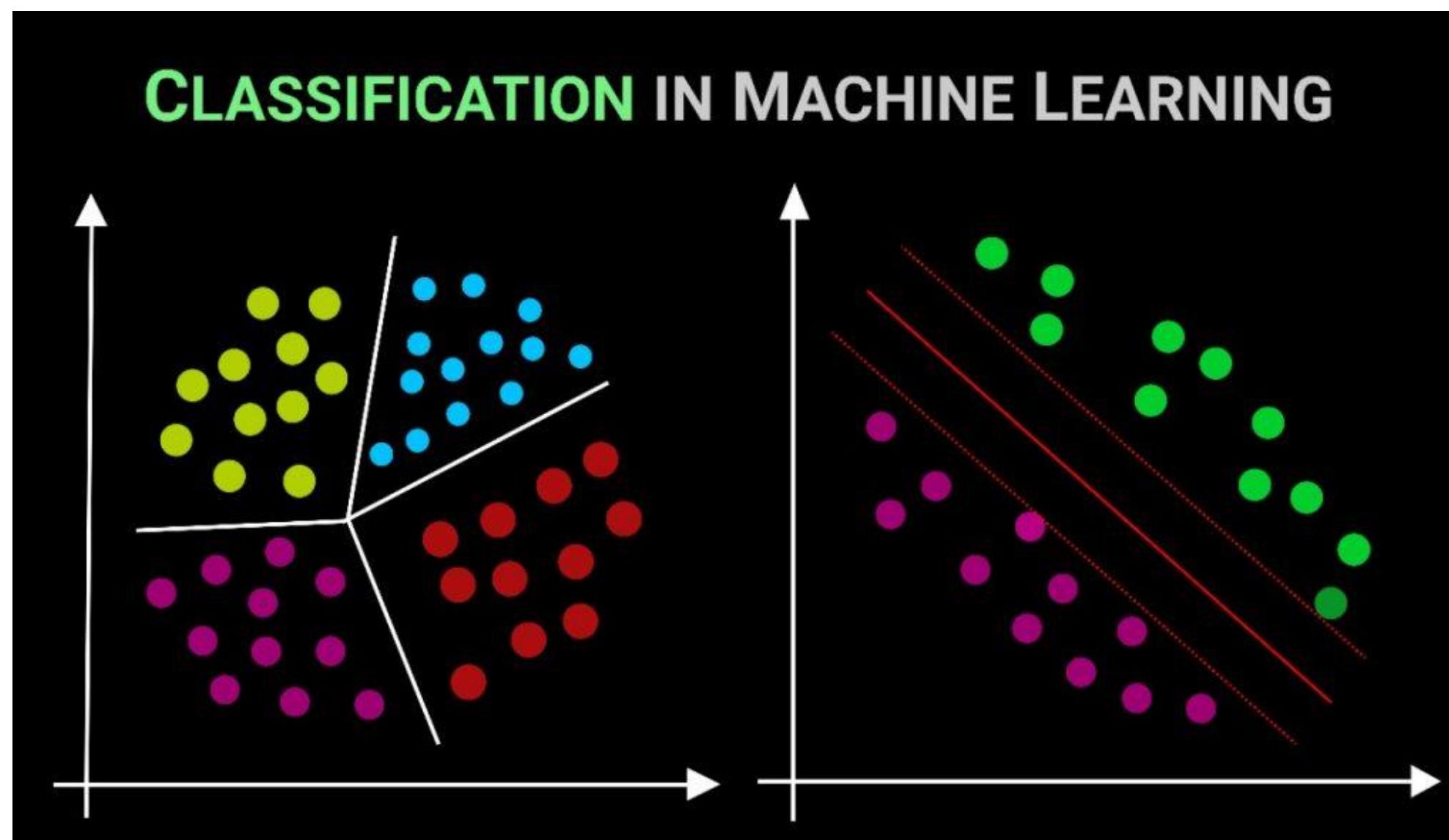
Machine Learning



ML Classifiers



ML Classifiers

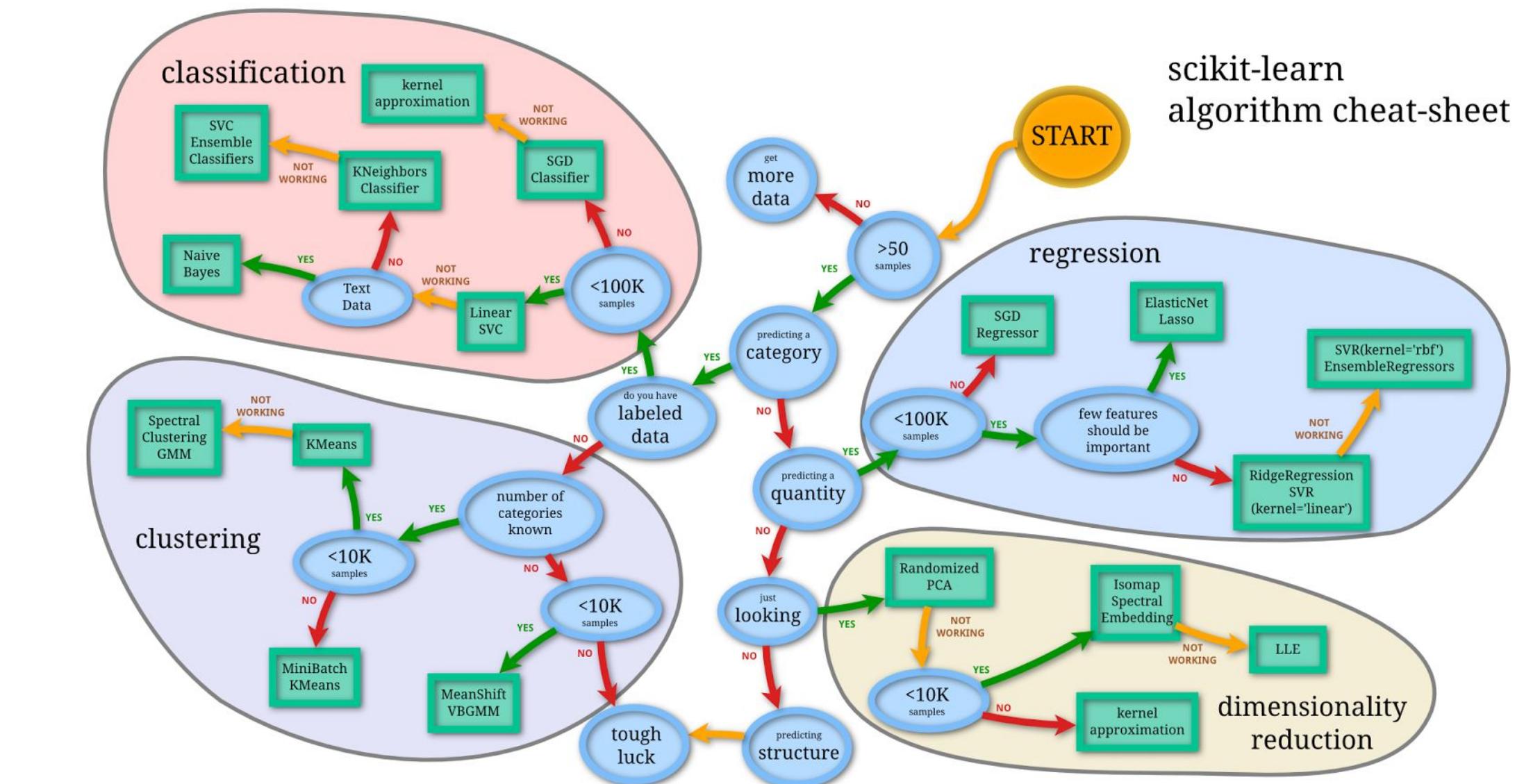


ML Classifiers

Google Earth Engine

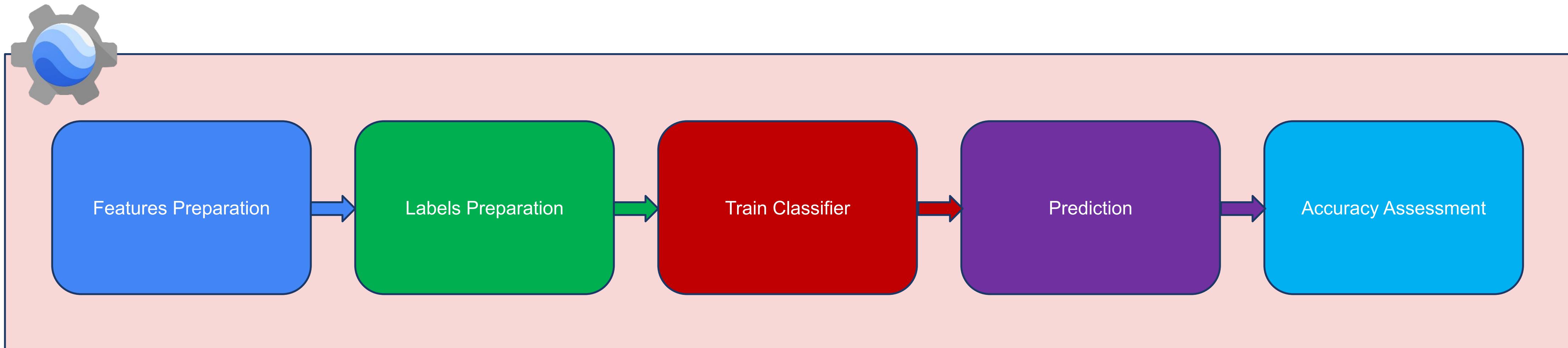
```
▼ ee.Classifier  
amnhMaxent  
confusionMatrix  
decisionTree  
decisionTreeEnsemble  
explain  
libsvm  
minimumDistance  
mode  
schema  
setOutputMode  
smileCart  
smileGradientTreeBoost  
smileKNN  
smileNaiveBayes  
smileRandomForest  
spectralRegion  
train
```

Scikit-learn



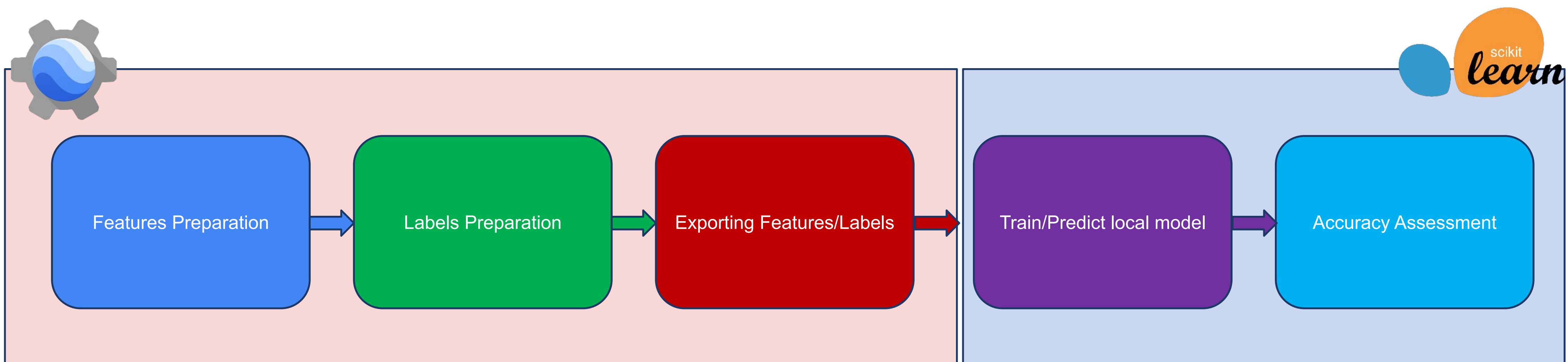
[Download VSCode](#)

Workflow 1: Classification in Earth Engine (geemap)



colab
[Open in Google Colab](#)

Workflow 2: Hybrid workflow (geemap + scikit-learn)



colab
[Open in Google Colab](#)



Ask Questions!



Mirza Waleed
PhD Fellow | GDE in Earth Engine
[@waleedgeo.com](http://waleedgeo.com)

