Logbook

Team Members: Matthew Kukucka, Jonathan Zou
Ann Arbor Huron High School: 12th grade
Project Name: TerraTracer

TOPIC RESEARCH

Identifying and Understanding:

What problem are you trying to solve? The more specific you are in describing the problem, the better your solution will be. How did you come up with the problem?

With the increasing rise in prominence of environmental research in the wake of the call to action from environmentalists, environmental trends have become increasingly important in the face of the new generations. Current environmental trends such as carbon footprints, ice cap levels, sea levels, and global temperature draw sharp comparisons to lesser known environmental trends such as carbon monoxide, nitrogen dioxide, and deforestation trends. While deforestation is a relatively well-known environmental trend, many people do not know the extent of the damaging effects. Since humans entered the scene of ecological damage, around 52% of the world's tropical forests have been cleared or degraded. From a global estimate, the World Wildlife Organization estimates that around 18 million acres of forest are lost per year globally, resulting in large CO2 emissions. WIth trees sequestering carbon over long periods of time before human intervention, carbon that is sequestered eventually is released into the atmosphere when they are cleared or not maintained. According to the National Geographic, more than 20% of global anthropogenic CO2 emissions are from deforestation; this accounts for more carbon release than the entire global transport sector, which only accounts for around 13%. To put this into proportion, a research study done by Seymour & Busch, 2016 with the World Resources Institute concluded that if "deforestation" were a country, it would rank third in global carbon emissions, only topped by China and the United States. With this devastating effect lying under humanity, awareness and further research must be brought around to examine and attempt to address this problem. With most data extraction being tedious and slow, algorithms through image processing are rising in prominence, which TerraTracer will look to develop from. Through our innovative product, TerraTracer, an supervised machine learning algorithm capable of analyzing terra vegetation indices, we will be able to accomplish a quick, efficient methodology in processing massive quantities of raw and previously analyzed data from databases such as the Google Earth Engine database to identify trends in vegetation for research, optimization, and awareness purposes.

Works Cited:

- Schoeneberger, Michele. (2009). Agroforestry: Working trees for sequestering carbon on agricultural lands. Agroforestry Systems. 75. 27-37. 10.1007/s10457-008-9123-8.
- Nunez, Christina. "Deforestation and Its Effect on the Planet." Deforestation Facts and Information, National Geographic LLC, 25 Feb. 2019, www.nationalgeographic.com/environment/global-warming/deforestation/.
- "Deforestation and Forest Degradation." WWF, World Wildlife Fund, www.worldwildlife.org/threats/deforestation-and-forest-degradation.
- "Deforestation and Forest Degradation." IUCN, International Union for Conservation of Nature, 5 Dec. 2018, www.iucn.org/resources/issues-briefs/deforestation-and-forest-degradation.
- "Tropical Deforestation." NASA, NASA, earthobservatory.nasa.gov/features/Deforestation/deforestation_update3.php
- Pimm, Stuart L. "Deforestation." Encyclopædia Britannica, Encyclopædia Britannica, Inc., 17 Dec. 2019, www.britannica.com/science/deforestation.
- Runyan, Christiane, and Paolo D'Odorico. Global Deforestation. Cambridge University Press, 2016.

- Gatehouse, Gabriel. “Deforested Parts of Amazon 'Emitting More CO2 than They Absorb'.” BBC News, BBC, 11 Feb. 2020, www.bbc.com/news/science-environment-51464694.

What is the result you are trying to achieve? The more specific you are in describing the result you want, the better your solution will be.

The results we are trying to achieve is a novel methodology in the analysis and characterization of terra vegetation through supervised machine learning. Our ultimate goal is for the machine learning algorithm to have the capability to analyze large amounts of raw and previously-analyzed data and create or confirm trends in terra vegetation through Python-based, regression-type ML using a variety of different resources and the Google Earth Engine database. The algorithm will serve to be a basis for research and awareness programs for universities and researchers. We hope our algorithm will have the capability to extend our regression-type ML to characterization as well as other regression types for other data indices other than terra vegetation in the future

Has this solution been done before?

In order to determine the feasibility and originality of our project, we had to examine current/existing solutions. The following research was conducted:

- Websites
    - Google Developer
        - Google Earth Engine Code Editor
    - Government Websites
        - NOAA, NASA, NCEP, NCAP
    - Machine Learning Websites
        - BuiltIn, TowardsDataScience, TopTal
    - Programming language
        - Python.org, Anaconda, Jupyter Notebook
- Database
    - Google Earth Engine Database
        - MODIS, Landsat, NCEP/NCAP Reanalysis, Sentinel-5P
    - Government Databases
        - NOAA, NASA, NCEP, NCAP
- Journals
    - Scientific
        - Nature, Scielo, MDPI, BMVC, ScienceDirect
- Programming/developer community network
    - Stack Overflow

Through this research, we identified four primary existing solutions that had positive attributes but ultimately were non-intrusive on our innovation. The following was found consistently across all existing solutions: there are very few linear regression-based machine learning algorithms (supervised)-- most algorithms are characterization-based. Existing algorithms mostly focus on characterizing agricultural data for different purposes from environmental. Most solutions are research papers on algorithms that are used by companies and freelance/independent farmers who are looking for sector growth specific boosting; many people use the existing solutions for growth optimization in industry rather than for awareness, research, or environmental purposes.

1. Strawberries
    a. Method for estimating leaf coverage in strawberry plants using digital image processing
        i. In this existing solution, researchers from Nueva Granada Military University examined the possibility of using ML as a conduit of digital image processing for strawberry fields. In the article, the researchers

cite that “the measurement of leaf coverage is considered as an exhaustive task for the researchers,” with the image identification and processing process being much more efficient. The methodology designed by the researchers is a non-destructive methodology with only the need for hundreds of data values from digital photographs. This research was focused on identifying the algorithm for agricultural purposes of growing strawberries and as a means of mass surveillance and prediction analyses for industrial managers as well as independent farmers. Its focus differs from TerraTracer.

   b. Sandino, Juan D., Ramos-Sandoval, Olga L., & Amaya-Hurtado, Darío. (2016). Method for estimating leaf coverage in strawberry plants using digital image processing. Revista Brasileira de Engenharia Agrícola e Ambiental, 20(8), 716-721. https://doi.org/10.1590/1807-1929/agriambi.v20n8p716-721

2. Maize
   a. Estimating Maize-Leaf Coverage in Field Conditions By Applying a Machine Learning Algorithm to UAV Remote Sensing Images
      i. In this existing solution, researchers from Zhejiang University in Hangzhou, China examined the potential use of UAV remote sensing images as training data for supervised ML algorithms in examining maize-leaf coverage and its growth. This is a characterization ML-based research article, immediately differing from our project’s purpose and goal. Using RGB-based technology with conventional equipment, the researchers concluded that the conventional equipment could be significantly reduced in favor of digital processing through image data. Plant phenotyping assumptions and conclusions laid out in the article solidify the article’s standing as an agricultural industry supplement, with which the algorithm can focus on. This fundamentally differs from TerraTracer’s goal.
   b. Zhou, C., Ye, H., Xu, Z., Hu, J., Shi, X., Hua, S., Yue, J., et al. (2019). Estimating Maize-Leaf Coverage in Field Conditions by Applying a Machine Learning Algorithm to UAV Remote Sensing Images. Applied Sciences, 9(11), 2389. MDPI AG. Retrieved from http://dx.doi.org/10.3390/app9112389
3. Soybean
   a. Soybean Leaf Coverage Estimation with Machine Learning and Thresholding Algorithms for Field Phenotyping
      i. In this existing solution, researchers from ETH Zurich examined estimating soybean leaf coverage with ML and thresholding algorithms. The focus of the algorithm and its combination with thresholding was to examine leaf coverage for soybean leaf identification. This is another characterization ML-based research article, immediately setting its purpose apart from our project. While there is nothing wrong with the research, the focus of the article fundamentally differs from our desired effect of TerraTracer since it is completely focused on soybean leaf coverage for agricultural use in predicting disease and growth (which is a different data index), allowing TerraTracer to stand on its own.
   b. Keller, K., Kirchgeßner, N., Khanna, R., Siegwart, R., Walter, A., & Aasen, H. (2018). Soybean Leaf Coverage Estimation with Machine Learning and Thresholding Algorithms for Field Phenotyping.
4. Tobacco
   a. Hyperspectral Imaging for Presymptomatic Detection of Tobacco Disease with Successive Projections Algorithm and Machine-learning Classifiers
      i. In this example, researchers from Zhejiang University in Hangzhou, China examined the potential of using hyperspectral imaging and unsupervised ML algorithms to characterize and classify the health of tobacco. The focus of the algorithm and its combination with the hyperspectral imaging was to introduce a non-intrusive successive projection analysis methodology that examined GCLM (grey-level co-occurrence matrix). However, while there is nothing wrong with the algorithm used in the research paper (as this is scientifically-validated research on one of the top research journals in the country), it involves the characterization of a specific type of vegetation--tobacco--that has little to no relevance to our project since it holds a different purpose.

b. Zhu, H., Chu, B., Zhang, C. et al. Hyperspectral Imaging for Presymptomatic Detection of Tobacco Disease with Successive Projections Algorithm and Machine-learning Classifiers. Sci Rep 7, 4125 (2017). https://doi.org/10.1038/s41598-017-04501-2

We will try to take all of the positive attributes of each of the existing solutions. Each product is scientifically-backed through journal publication on multiple different scientific publications. We will be looking for scientifically-valid data through the Google Earth Engine database, which is raw data from satellites. We will also be implementing the training to test data ratio used by most of the machine learning algorithms in the different articles since it is independently verified by multiple different sources--universities and researchers.

Ideating:

What are some possible solutions? Which one did you choose to pursue? How did you decide which solution to try? The more specific you are in describing the solution you will create, the better your invention will be. How did you come up with the solution?

Decision Matrix for Algorithm/Dataset Selection

Dataset Selection (1-100)	Extent of Data (1 - Lack of data, 100 - Maximum Data extent possible	Visualization (1 - Least visually appealing/presentable, 100 - Maximized visual appeal/presentation	Environmental Relevancy (1 - Least effect on environment/not relevant, 100 - massive effect on environment)	Algorithm Difficulty/Feasibility (1 - Most Difficult, 100 - Easier)	Average (average of parameter ratings)
Terra Vegetation (Design Sketch #1)	**28**	**100**	**50**	**60**	**59.5**
Nitrogen Dioxide (Design Sketch #2)	4	80	70	60	53.5
Carbon Monoxide (Design Sketch #3)	4	80	70	60	53.5
Water Vapor (Design Sketch #4)	100	10	30	30	42.5
Surface Reflectance (Design Sketch #5)	29	30	30*	10	24.75

Explanation:

There are **five** main datasets that can be selected-- terra vegetation, nitrogen dioxide, carbon monoxide, water vapor, and surface reflectance. Each of these datasets are compiled with data from different satellites as dictated below (with the exception of the NCEP/NCAR Reanalysis). The algorithm design sketches are relatively similar to each other, making the overall selection of which dataset to use the most important decision making process in need of a unique decision matrix.

- Terra Vegetation
    - "MOD13Q1.006 Terra Vegetation Indices 16-Day Global 250m." *Google Earth Engine*, NASA LP DAAC at the USGS EROS Center, 18 Feb. 2000, developers.google.com/earth-engine/datasets/catalog/MODIS_006_MOD13Q1.
- Nitrogen Dioxide
    - "Sentinel-5P NRTI NO2: Near Real-Time Nitrogen Dioxide." *Google Earth Engine*, European Union/ESA/Copernicus, 10 July 2018, developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_NRTI_L3_NO2.
- Carbon Monoxide
    - "Sentinel-5P NRTI CO: Near Real-Time Carbon Monoxide." *Google Earth Engine*, European Union/ESA/Copernicus, 22 Nov. 2018, developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_NRTI_L3_CO.
- Water vapor
    - "NCEP/NCAR Reanalysis Data, Water Vapor." *Google Earth Engine*, NCEP/NCAR, 1 Jan. 1948, developers.google.com/earth-engine/datasets/catalog/NCEP_RE_surface_wv .
- Surface Reflectance
    - "Landsat 7." *Google Earth Engine*, USGS/NASA, 1 Jan. 1999, developers.google.com/earth-engine/datasets/catalog/landsat-7/ .

Dataset Selection Variables:

1. **Terra Vegetation**: "MOD13Q1.006 Terra Vegetation Indices 16-Day Global 250m." *Google Earth Engine*, NASA LP DAAC at the USGS EROS Center, 18 Feb. 2000, developers.google.com/earth-engine/datasets/catalog/MODIS_006_MOD13Q1.
    a. Dataset Satellite: MODIS
    b. Dataset Availability: 2000-present
    c. Dataset Objective: Normalized difference vegetation index (NDVI) and vegetation index (VI) of terra vegetation
2. **Nitrogen Dioxide**: "Sentinel-5P NRTI NO2: Near Real-Time Nitrogen Dioxide." *Google Earth Engine*, European Union/ESA/Copernicus, 10 July 2018, developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_NRTI_L3_NO2.
    a. Dataset Satellite: Copernicus Sentinel-5 Precursor
    b. Dataset Availability: 2017-present
    c. Dataset Objective: Nitrogen Dioxide emissions/levels (ppm)
3. **Carbon Monoxide**: "Sentinel-5P NRTI CO: Near Real-Time Carbon Monoxide." *Google Earth Engine*, European Union/ESA/Copernicus, 22 Nov. 2018, developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_NRTI_L3_CO.
    a. Dataset Satellite: Copernicus Sentinel-5 Precursor
    b. Dataset Availability: 2017-present
    c. Dataset Objective: Carbon Monoxide emissions/levels (ppm)
4. **Water Vapor**: "NCEP/NCAR Reanalysis Data, Water Vapor." *Google Earth Engine*, NCEP/NCAR, 1 Jan. 1948, developers.google.com/earth-engine/datasets/catalog/NCEP_RE_surface_wv .
    a. Dataset "Satellite": NCEP/NCAR Reanalysis I
    b. Dataset Availability: 1948 to present
    c. Dataset Objective: Water vapor tracking concentration logging
5. **Surface Reflectance**: "Landsat 7." *Google Earth Engine*, USGS/NASA, 1 Jan. 1999, developers.google.com/earth-engine/datasets/catalog/landsat-7/ .
    a. Dataset Satellite: Landsat 7

b. Dataset Availability: 1999-present
c. Dataset Objective: Near/short-wave infrared bands to assess surface reflectance and brightness temperature

Design Matrix Parameters/Designations:

Our decision matrix includes multiple design matrix parameters that are used to judge the final design chosen. These parameters are specifically chosen to optimize the feasibility and complexity at the same time in pursuing the algorithm generation. Our project includes several design matrices throughout the decision making process. The first of which is used to determine the dataset to be used and analyzed. In order to determine the dataset to be evaluated for algorithm development, our team used four main criteria to judge the datasets.

1. Extent of Data
    a. The extent of the data is the most important factor when it comes to selecting a dataset. With a supervised ML algorithm, the algorithm, while not requiring as much data as unsupervised machine learning algorithms, requires a medium to large dataset in order to analyze it. With most satellite types (Landsat 7, MODIS, Sentinel-5P) taking pictures on 8-day indices, the extent, or the amount of data, can be correlated with the amount of time a satellite has been in orbit. Our criteria for extent of data was evaluated using a simple proportion: using the oldest dataset available, NCEP/NCAR Reanalysis I Data, dating back from 1948 to present (72 years), we constructed a simple year by year proportion rating for each dataset.
        i. Terra Vegetation- MODIS (2000-present) → 20 years → 20/72 years = 0.28
        ii. Nitrogen Dioxide- Sentinel-5P (2017-present) → 3 years → 3/72 years = 0.04
        iii. Carbon Monoxide- Sentinel-5P (2017-present) → 3 years → 3/72 years = 0.04
        iv. Water Vapor- NCEP/NCAR Reanalysis I (1948-present) → 72 years → 72/72 years = 1
        v. Surface Reflectance- Landsat 7 (1999-present) → 21 years → 21/72 years = 0.29
2. Visualization
    a. The visualization of the data is also an important factor when it comes to selecting a dataset. There is inevitably going to be a need for a visual aspect of our algorithm→ this will most likely be accomplished throughout a set of graphical analyses, 3D physical topographical maps, and potential time lapses of datasets. Visualization was based off of map resolution as well as ease of understanding. Note that visualization is also heavily dependent on the type of satellite used to collect data for the dataset-- satellites vary in resolution as well as frame indices.
        i. Terra Vegetation- Clear, 1 kilometer - 250 meter spatial resolution setting → enables users to clearly see the global trend of vegetation from afar but maintains resolution (due to 1km spatial resolution) when zooming on certain selected areas = 100
        ii. Nitrogen Dioxide- Clear, 500 meter spatial resolution setting → enables users to clearly see the global trend of nitrogen dioxide from afar but begins to lose resolution (due to 500 meter spatial resolution) when zooming on certain selected areas - approximately ⅘ of resolution of terra vegetation = 80
        iii. Carbon Monoxide- Clear, 500 meter spatial resolution setting → enables users to clearly see the global trend of carbon monoxide from afar but begins to lose resolution (due to 500 meter spatial resolution) when zooming on certain selected areas - approximately ⅘ of resolution of terra vegetation = 80
        iv. Water vapor- Unclear, blocked, heavily pixelated due to massive amount of data, 6-hour temporal resolution and 2.5 degree spatial resolution combine historical data and current atmospheric state→ makes data extremely confusing and blown out of proportion in visualization = 10
        v. Surface Reflectance- Limited visualization-- uses bands instead of spatial res. Cloud, shadow, water, and snow masks are produced using CFMASK-- lack of cloud reduction makes visualization difficult but extremely so like water vapor = 30
3. Environmental Relevancy
    a. The relevancy of the dataset and the data it pertains to is extremely important. With the direction of the project focused on the characterization of an environmental trend, the environmental trend has to be relevant-- a statistically significant impact-- on the environment in order to be considered for use.

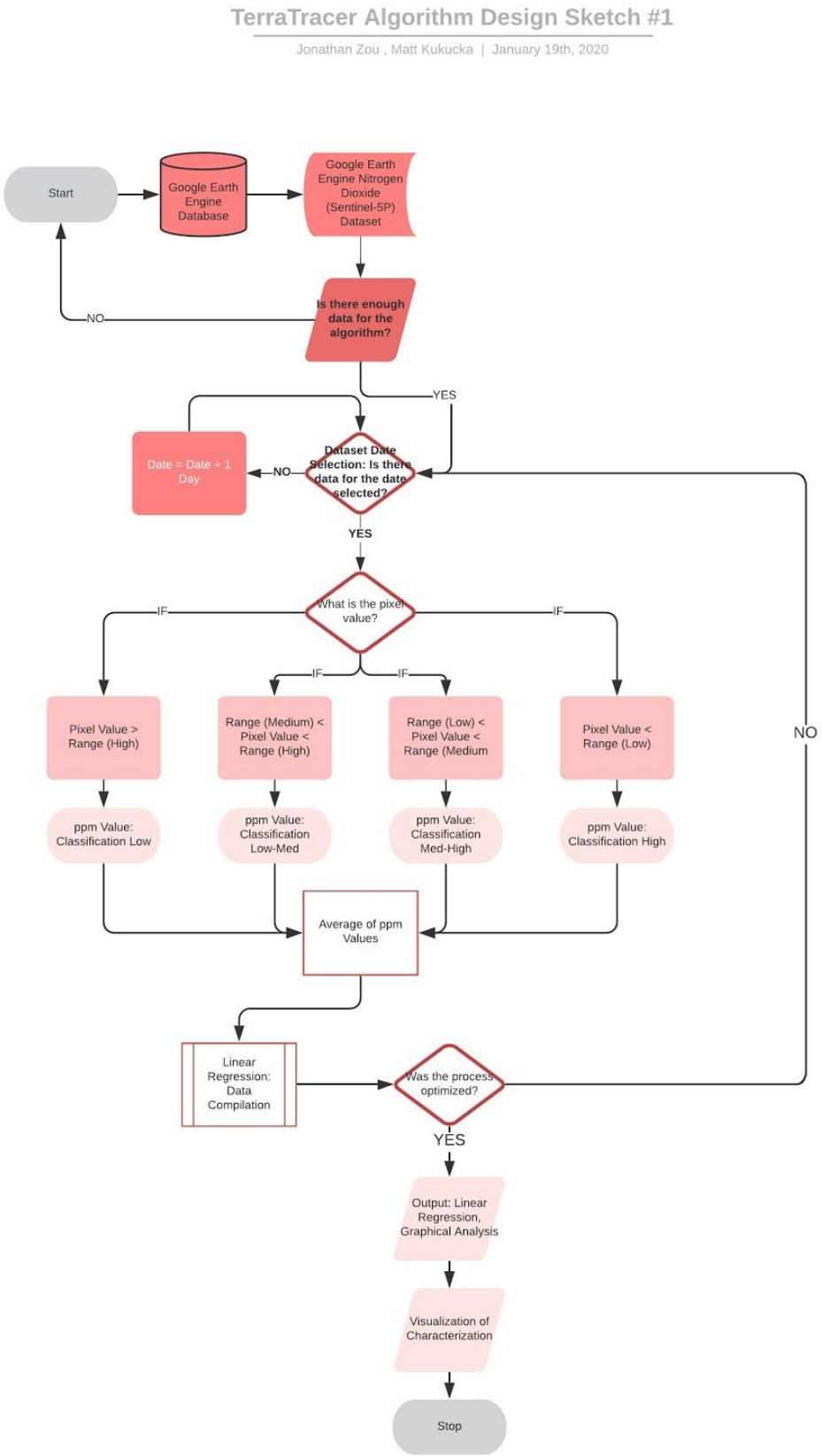
i. Terra Vegetation- Terra vegetation is affected by seasons (can be counteracted by only examining spring/summer seasons)-- vegetation can be correlated with global warming, deforestation, and a number of different human-led causes of environmental damage = 50
ii. Nitrogen dioxide- Greenhouse gas, directly contributes to global warming, produced from motor vehicle exhaust (around 80%), extremely relevant = 70
iii. Carbon monoxide- Greenhouse gas, directly contributes to global warming, produced from fuel burning (stoves, motor vehicles, etc.), extremely relevant = 70
iv. Water Vapor- Greenhouse gas, occurs naturally, hard to control or regulate makes it less relevant = 30
v. Surface Reflectance- Surface reflectance can be correlated with the amount of vegetation, the amount of ice build-up, the amount of precipitation, the amount of human urbanization, etc.-- however, it, in in of itself, is not directly positively correlated with global warming-- it is simply an indicator, less relevant = 30

4. Algorithm Difficulty/Feasibility
    a. The difficulty and feasibility behind the actual design of the algorithm is important. With a supervised ML program, this is directly correlated with the amount of data there is to work with as well as the type of algorithm that is being run. In our case, we used a 100 point scale from 0 to 100 with 0 behind the hardest and 100 being the easiest. Our team wants to focus on the end product of the algorithm, hence the emphasis on an easier, more feasible, yet functional algorithm.
        i. Terra vegetation- Algorithm based off of pixel identification in image classification = 60
            1. ML Concepts: pixel identification, image classification, ML optimization
        ii. Nitrogen dioxide- Algorithm based off of pixel identification in image classification = 60
            1. ML Concepts: pixel identification, image classification, ML optimization
        iii. Carbon monoxide- Algorithm based off of pixel identification in image classification = 60
            1. ML Concepts: pixel identification, image classification, ML optimization
        iv. Water vapor- Algorithm based off of pixel identification in images, pixels are more separated, making them easier to identify, however→ the data collected for water vapor is not restricted to one satellite like the other datasets: the NCEP/NCAR Reanalysis draws from multiple satellites (ERBE, CERES, ISCCP, etc.) as well as surface data from environmental flagships and ground data reporting = 30
            1. ML Concepts: pixel identification, image classification, ML optimization, dataset manipulation (multiple source dataset analysis)
        v. Surface Reflectance- Algorithm based off of pixel identification in images, however, images used are processed, with data missing and cloud coverage interference-- algorithm would have to incorporate cloud interference logging and analysis (see design sketch #5 above), which makes the algorithm significantly more difficult to write = 10
            1. ML Concepts: pixel identification, image classification, ML optimization, data mining (cloud coverage interference), cloud logging and analysis

Spatial Resolution Decision

This decision doesn't require a decision matrix because there is only one constraint, sample size. With that being said, the Terra Vegetation indices dataset offers 3 different spatial resolutions: 250 m, 500 m, and 1 km. Spatial resolution refers to the number of pixels utilized in the construction of an image, or in radiography, the ability of an imaging system to differentiate between two near-by objects. Having a lower spatial resolution will cause more data to be grouped in a large quantity in one pixel. Not only does this reduce the sample set size in general, but it also makes the dataset less precise for the area it's representing in that pixel. Therefore, having a higher pixel density will allow for the algorithm to analyze a larger amount of precise data, so we choose to use the 250 m spatial resolution dataset.

Draw a model (a sketch or drawing) of the invention you are thinking about building. Label all the important parts and features.



What problems or issues might you encounter with this design? Is this design compatible with the principle of sustainability? Who did you talk to about this design (another student, parent, teacher, etc.)?What were their comments about your design?

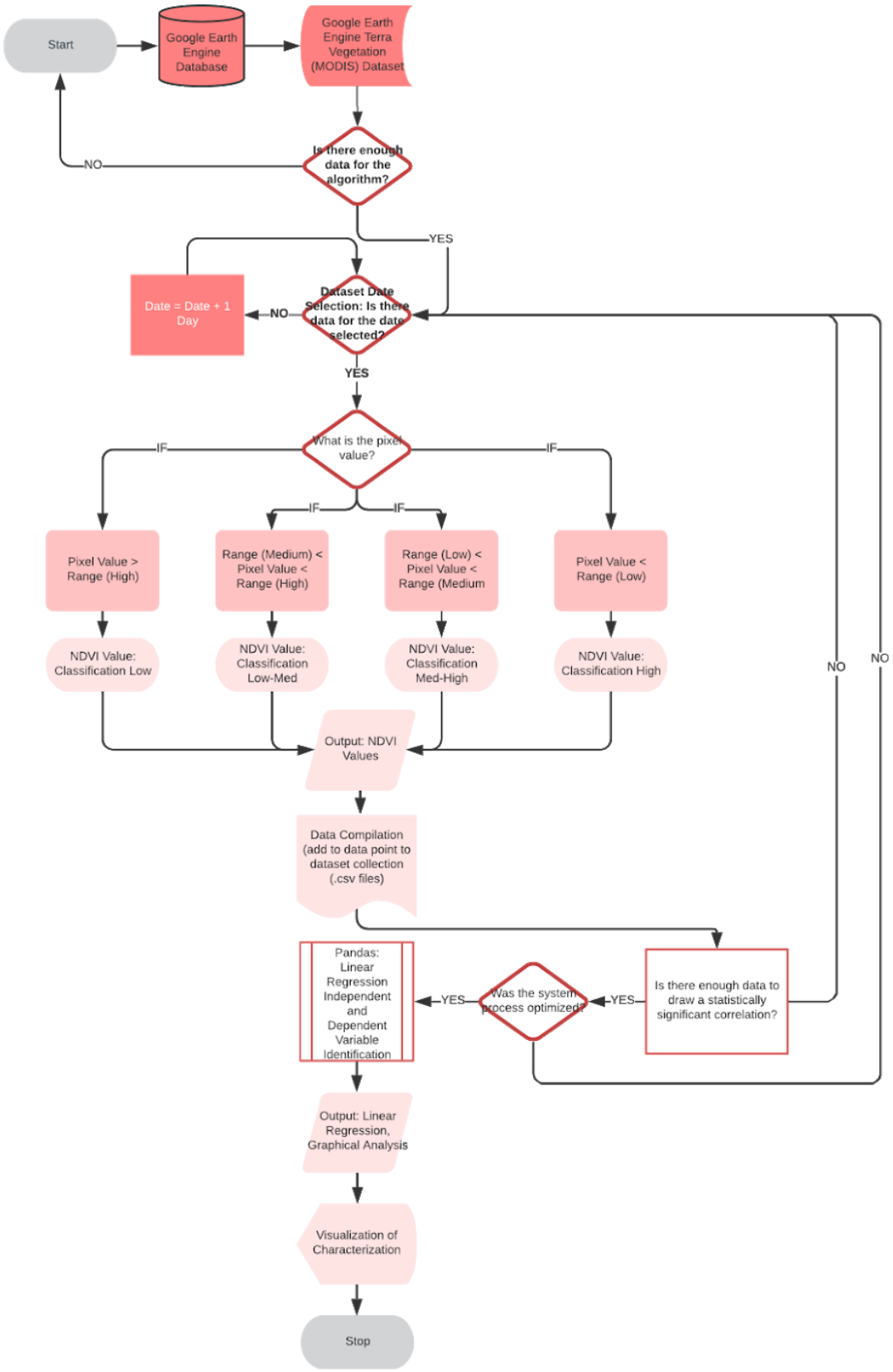
There are multiple problems and issues we could encounter in an attempt to design and validate our design; however, these problems can be addressed very thoroughly with proper research and consumer feedback. Problems over originality, the smoothness of the algorithm's operation, the data extraction from Google Earth Engine, and the design of the algorithm itself and its functionality in carrying out its base purpose could be complicated and riddled with issues. These problems will inevitably arise as coding an algorithm requires multiple rounds of testing and refining the data ratio as well as the code itself to fix the problems at hand. TerrTracer is definitely compatible with the principle of sustainability; not only does it use only open-source, web-based resources, but it also it an algorithm focused on bringing awareness to sustainability in the division of efficient land use and sustainable development. We consulted with Mr. Cupit, our engineering teacher, about the idea of writing an algorithm to set trend prediction and identification; he believed it would be useful to have a time lapse of the succession of algorithm inputs and outputs as it "trained" itself, something which we have now included into our goals for the project. We will also glean feedback from interacting with computer science students at the University of Michigan, regular people who have little knowledge of ML for the aesthetic and visual appeal, and other math and computer science teachers at our high school.

How can you fix those problems or address those issues?

Fixing the aforementioned problems is difficult to identify before conducting the testing. As mentioned before, Problems over originality, the smoothness of the algorithm's operation, the data extraction from Google Earth Engine, and the design of the algorithm itself and its functionality in carrying out its base purpose could be complicated and riddled with issues; however, this can be addressed through thorough testing and refining in multiple stages to ensure that the issues are resolved to their optimum capacity. With proper research, data extraction, and feedback from experts, TerraTracer should go through multiple levels of refining before the final design is revealed-- however, with the following steps taken, we can minimize the amount of errors and issues that we encounter by completing the work and research ahead of time.

Repeat steps 5 to 7 until you have a design that you think will work. You may have to make multiple copies of a blank page until you have a good design.

Final Design:



What parts, materials, and tools will you need to make the invention and how much will they cost?

1. Hardware
    a. Two sets of laptops/desktops capable of running (already in possession)
        i. Supervised machine learning algorithm from Python notebook
        ii. Google Earth Engine visualization
        iii. Video animation
    b. Potential 3D topographical map (varies on scale, but will cost money -- TBD)
2. Software
    a. Google Earth Engine (free)
        i. Datasets
        ii. Google Earth Engine Code Editor
        iii. Visualization
    b. Python Notebook (free)
        i. Project Jupyter
        ii. Anaconda/Miniconda
        iii. Google Colab Notebook
    c. Time Lapse/Video Editing (free)
        i. Lightworks, Adobe, Prezi
    d. Data characterization software (free)
        i. Minitab
        ii. Google Sheets/Microsoft Spreadsheets

Most materials required are either free or already in possession by the team, except for the potential 3D printed topographic map. Without taking that into consideration, the project won't cost us anything since everything is open-source.

Where will you get those parts and materials?

Due to the project being mainly based on open-source software and programs, we will be getting these parts and materials from open-source libraries as well as from online sources. With the use of open-source resources, we have the capability to be completely sustainable in the development of our project.

What additional skills or abilities will you need to make the invention?

We will need to be proficient in Python Notebook (Anaconda/Miniconda), Google Colab Notebook, video editing, data characterization software, Python programming language, Google Code Editor, Google Earth Engine. Both members of our group will also look to take machine learning college crash courses to better and refresh our knowledge of our project-- the skills obtained from the courses will work side by side with our problem identification and issue refinement throughout the course of the project, especially when it comes to resolving and fixing problems that require modifications and improvements.

Who can help you build the invention?

As we are situated in the backyard of the University of Michigan, we can receive direction from computer science/statistics students, graduate students, and professors, who can give vital insight into the data analysis process, especially in regards to the python machine learning methodology. Additionally, we can refer to math/statistics teachers at Huron who can provide a greater understanding behind the statistical relevance of our collected data.

Constraints:

We decided to streamline our design and project process with the following constraints-- this provided us a structure to follow throughout our process:

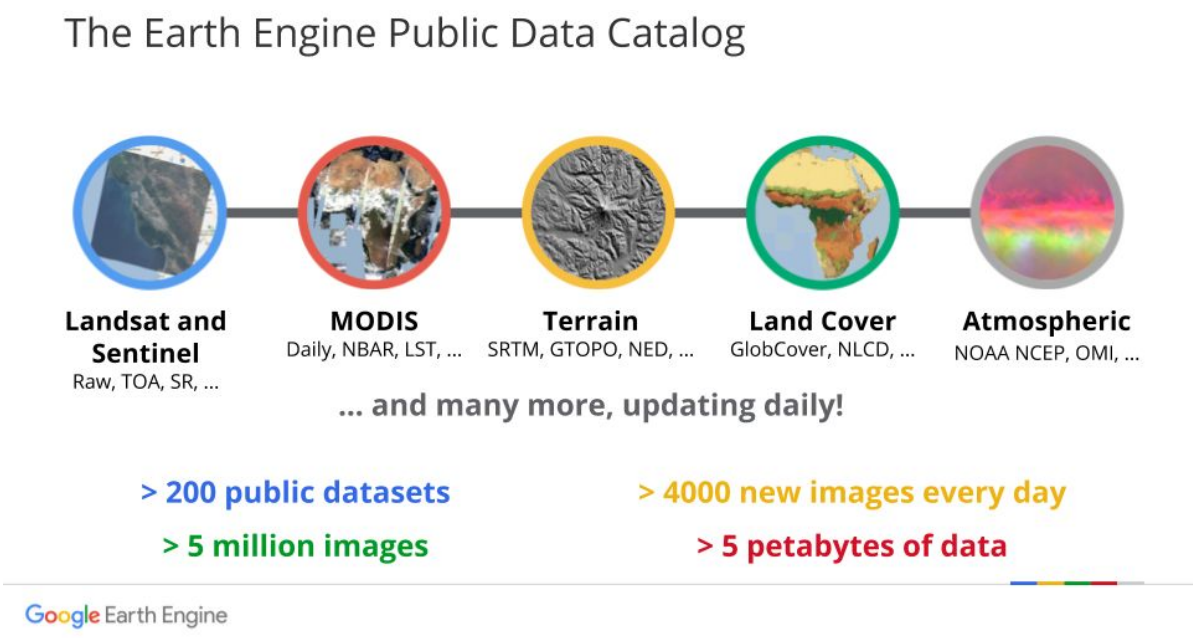
a. Create a functional, supervised machine learning algorithm that is capable of representing Google Earth Engine's terra vegetation indices dataset.
    i. This algorithm's functionality is determined by its capability to receive and analyze the dataset (of pixel spatial resolution or density) to output trends (or regressions) from the data.
    ii. A key characteristic of this ML algorithm is that it'll be able to optimize its analyzations to improve its performance with more "training", much like how a human becomes more efficient at a task with experience.
b. The algorithm will be able to make future predictions on the direction of terra vegetation density based off of the given dataset's trends.
c. The algorithm will be coded using Python 3.7 programming language.
d. Dataset is derived from Google Earth Engine.
    i. MOD13Q1.006 Terra Vegetation Indices 16-Day Global 250m
e. Create a video detailing the processes behind the algorithm in order for it to identify trends and make predictions.

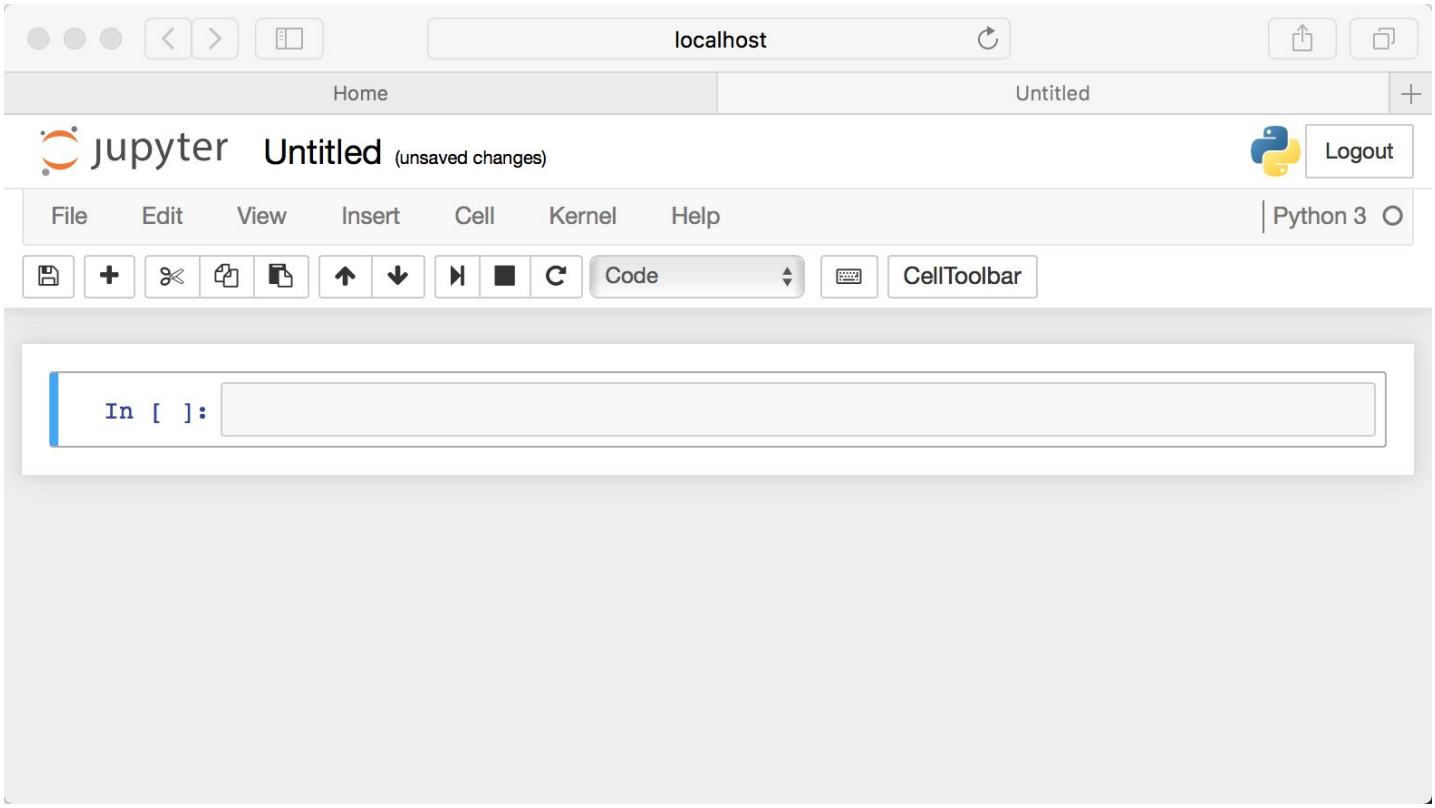
CONSTRUCTION/TESTING

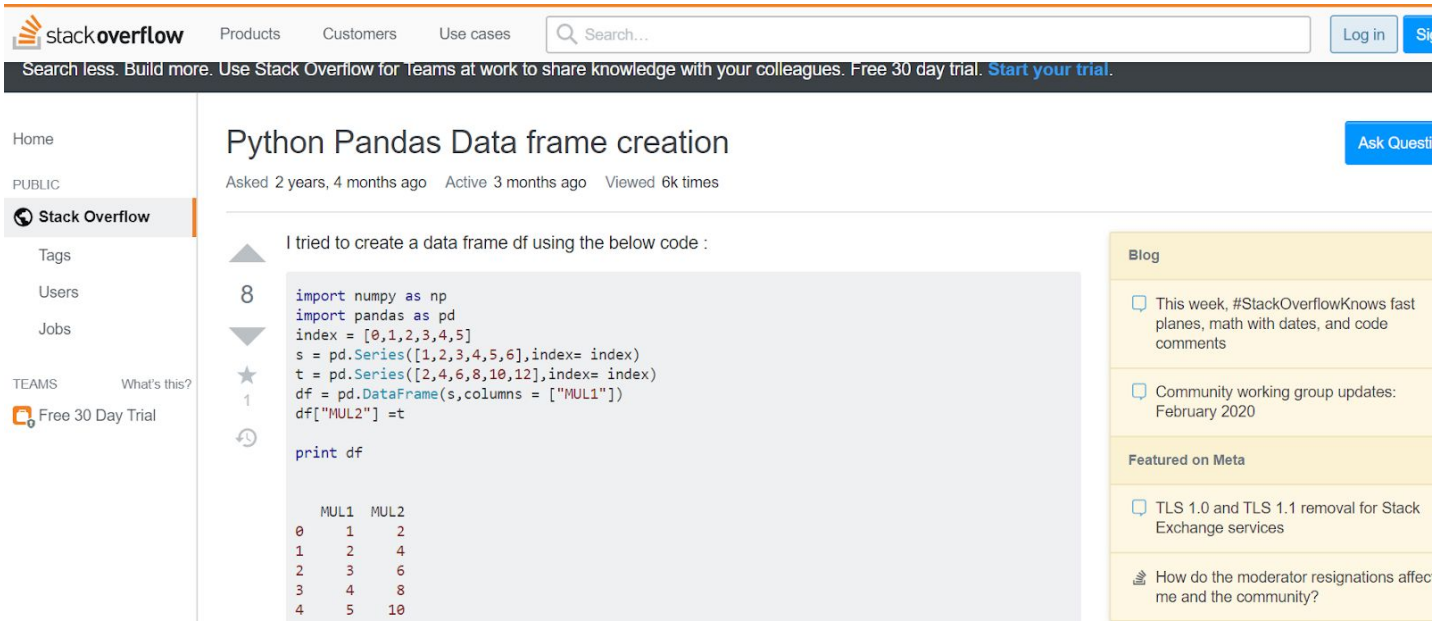
Designing, Building, Testing and Refining:

A key step to any project is documenting the process of designing (as mentioned in the previous sections), building, testing and refining. The iterations as well as the corresponding images reveal the construction (building) process of the entire project, the testing phases of each iteration, and the refining (improvements and modifications made to the design and direction of the project during the project timeline):

Iteration 1







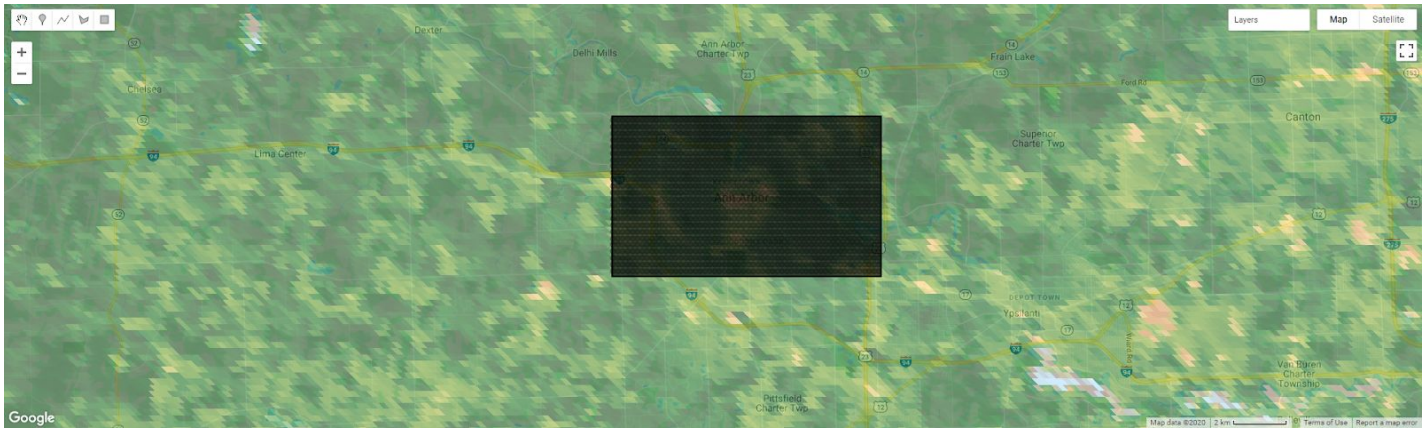
The first iteration of this project mainly involved our familiarization with the resources we will be utilizing throughout the project. One of such resources is Google Earth Engine, including the satellite data collections. Google Earth Engine is a JavaScript API, and with the code editor to develop algorithms that will be able to analyze the data indices (in our case, MODIS terra vegetation 250m) graphically and numerically. In order to properly use Google Earth Engine, we will need to know how to code in Java and have an understanding of the dataset (what does an NDVI show, how is it represented as a map layer, etc.). Our understanding will need to cover data extraction using a point reduction method. Additionally, we will be utilizing Jupyter Notebook, an open-source programming environment, which will contain our code for machine learning data analysis after extraction. The code will be done with Python 3.7, a high-level programming language, so we will need to supplement our familiarization of Jupyter Notebook with StackOverflow, a resource which contains questions/answers on programming. These two resources used in conjunction will allow us to develop a greater understanding of Python, eventually leading to us able to implement a machine learning algorithm to predict future data based on extracted data.

Iteration 2

Google Earth Engine Data Extraction Code

```
//Display Terra Vegetation Layer
var dataset = ee.ImageCollection('MODIS/006/MOD13Q1')
                .filter(ee.Filter.date('2010-04-01', '2010-06-30'));
var ndvi = dataset.select('NDVI');
var ndviVis = {
  min: 0.0,
  max: 10000.0,
  palette: [
    'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
    '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
    '012E01', '011D01', '011301'
  ],
};
Map.addLayer(ndvi, ndviVis, 'NDVI');
//Set data collection zone
var rectangle = ee.Geometry.Rectangle(-68.96, -1.77, -57.27, -7.12);
Map.centerObject(rectangle);
Map.addLayer(rectangle, {}, 'rectangle')
var dictionary = ee.Image.pixelLonLat().reduceRegion({
  reducer: ee.Reducer.toCollection(['longitude', 'latitude']),
  geometry: rectangle,
  scale: 250
});
var points = ee.FeatureCollection(dictionary.get('features'))
    .map(function(feature) {
      var lon = feature.get('longitude');
      var lat = feature.get('latitude');
      return ee.Feature(ee.Geometry.Point([lon, lat]), {
        'featureID': ee.Number(lon).multiply(1000).round().format('%5.0f')
            .cat('_')
            .cat(ee.Number(lat).multiply(1000).round().format('%5.0f'))
      });
    });
print('points', points)
Map.addLayer(points);
var dataset = ee.ImageCollection("MODIS/006/MOD13Q1")
  .filterDate('2010-04-01', '2010-06-30')
  .select('NDVI')
print('dataset', dataset)
var triplets = dataset.map(function(image) {
  return image.reduceRegions({
    collection: points,
    reducer: ee.Reducer.first().setOutputs(image.bandNames()),
    scale: 250,
  }).map(function(feature) {
    return feature.set({
```

```
      'imageID': image.id(),
      'timeMillis': image.get('system:time_start')
    })
  });
}).flatten();
print(triplets)
var format = function(table, rowId, colId, rowProperty, colProperty) {
  var rows = table.distinct(rowId);
  var joined = ee.Join.saveAll('matches').apply({
    primary: rows,
    secondary: table,
    condition: ee.Filter.equals({
      leftField: rowId,
      rightField: rowId
    })
  });
  return joined.map(function(row) {
    var values = ee.List(row.get('matches'))
      .map(function(feature) {
        feature = ee.Feature(feature);
        return [feature.get(colId), feature.get(colProperty)];
      }).flatten();
    return row.select([rowId, rowProperty]).set(ee.Dictionary(values));
  });
};
var results = format(triplets, 'imageID', 'featureID', 'timeMillis', 'NDVI');
print(results)
// Note that there's a dummy feature in there for the points ('null').
var transpose = format(triplets, 'featureID', 'imageID', 'null', 'NDVI');
print(transpose)
Export.table.toDrive({
  collection: results,
  description: 'data',
  fileNamePrefix: 'data',
  fileFormat: 'CSV'
});
```



The code above makes up the data extraction process. The language of the code is javascript. This process is run in a predetermined export region, in this case, a rectangle. The extraction process will only operate in this region. The process is able to extract the data by overlaying reduction points, that is, points that are assigned to each spatial pixel (remember, 250 m in length) within the region, extracting the correlating NDVI value from that pixel. Then, the program organizes the time stamp, coordinate point, and NDVI value into a CSV spreadsheet (comma separated values) which allows the algorithm to analyze the trends across time. The export region is shown above, with the reduction points also included inside.

Now, we will detail the analysis algorithm itself, coded in Python 3.7 on Jupyter Notebook. When referencing "cells", we are talking about the blocks of code which are organized together. The leftmost column has the name of the cells.

Dataset

In [124]:
```
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
import numpy as np
import datetime
```

In [125]:
```
url = r"C:\Users\mattk\Desktop\master - master.csv"
dataset = read_csv(url)
```

In [126]:
```
dataset = dataset.rename(columns={"system:index": "Date"})
```

In [127]:
```
dataset = dataset.replace(np.NaN, 0)
```

In [128]:
```
dataset
```

Out[128]:

	Date	125001_40200	125001_40289	125090_40020	125090_40110	125090_40200	125090_40289	125090_40379	125180_39930	125180_40020	...
0	2001_12_03_0	6516	4778	3937.0	3615	4022	3600	2641	1985	3128	...
1	2001_12_19_0	5637	3892	4300.0	3196	3111	2801	2478	1871	3434	...
2	2002_01_01_0	3178	3570	1461.0	2757	1136	1061	215	64	1314	...
3	2002_01_17_0	6179	4641	5019.0	2980	2863	2429	2505	371	1950	...
4	2002_02_02_0	5008	4221	4030.0	2954	3345	2768	2299	1614	3188	...
...	...	...	...	...	...	...	...	...	...	...	...
83	2012_01_17_0	3812	4549	3606.0	3382	3263	2949	2124	2002	3488	...
84	2012_02_02_0	3632	4107	3293.0	2196	2987	2715	2183	1820	3019	...
85	2012_02_18_0	3633	3905	3280.0	2867	2738	2907	2031	1971	2864	...
86	2012_03_05_0	3196	4145	3144.0	3103	3094	3184	2070	2044	3594	...
87	2012_03_21_0	3577	3952	2753.0	3101	3292	3371	2328	2115	3390	...

88 rows × 491 columns

In cell [124], we are merely importing software libraries written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
Cell [125] establishes what CSV spreadsheet to use as the dataset in its analysis.
Cell [126] renames the timestamp column from "system:index" to "Date". This is an unnecessary edit, but it makes the table more organized.
Cell [127] replaces all unregistered NDVI values with 0. Sometimes the satellite is unable to determine the index value at a particular point, so it assigns it a NaN value (Not a Number). This proved to make data representation difficult (principally graphical analysis and array calculations), so by replacing it with a 0, it alleviates this issue.
Cell [128] shows the organized dataset in an organized table, with rows representing timestamps and columns representing the value's coordinate location.

In [129]:
```
dates = []
values = []
for ind, row in dataset.iterrows():
    row_x = row
    dates.append(row_x[0])
    values_date = row_x[1: -1]
    values.append(values_date)
```

In [130]:
```
values = np.asarray(values)
values
```

Out[130]:
```
array([[6516, 4778, 3937.0, ..., 2788.0, 3685.0, 3139.0],
       [5637, 3892, 4300.0, ..., 2846.0, 3622.0, 3715.0],
       [3178, 3570, 1461.0, ..., 471.0, 2466.0, 2115.0],
       ...,
       [3633, 3905, 3280.0, ..., 1979.0, 2820.0, 0.0],
       [3196, 4145, 3144.0, ..., 503.0, 2928.0, 0.0],
       [3577, 3952, 2753.0, ..., 2476.0, 2967.0, 0.0]], dtype=object)
```

In [131]:
```
def convert_date(dates):
    # loop over entries in dates, convert each date to integer that is number of days after day 0
    # add this converted date to an array called "converted_dates, iterate over all"
    converted_dates = []
    start_date = dates[0]
    start_year = int(start_date[:4])
    start_month = int(start_date[6:7])
    start_day = int(start_date[8:10])
    a = datetime.datetime(start_year, start_month, start_day, 0, 0, 0)
    for i in range(len(dates)):
        date  = dates[i]
        year = int(date[:4])
        month = int(date[5:7])
        day = int(date[8:10])
        print(day)
        b = datetime.datetime(year, month, day, 0, 0, 0)
        time_difference = b-a
        converted_difference = time_difference.days
        converted_dates.append(converted_difference)
    return converted_dates #return converted_dates
```

Cell [129] establishes two variables: dates and values. This is an important step in the process of plotting the values later on.
Cell [130] equates the values array to an array of the NDVI values per row. This array is a 1 row matrix containing all the individual NDVI values for a given timestamp in a row.
Cell [131] is a very necessary element of the data analysis process. The timestamps prior to conversion are represented in strings, which makes it impossible to plot due to the underscore. This cell converts all of the string dates into integer values, which makes the data graphical now.

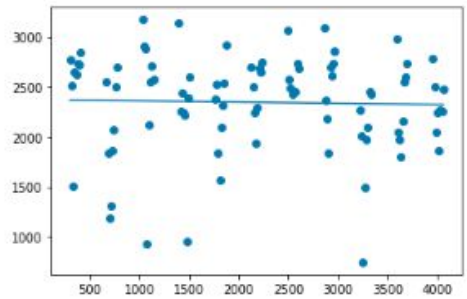
In [134]:
```
import matplotlib.pyplot as plt
```

In [135]:
```
duplicate_dates = []
for i in range(len(converted)):
    length = len(values[i])
    duplicate_date = converted[i]*np.ones(length)
    duplicate_dates.append(duplicate_date)
```

In [137]:
```
#duplicate_dates_1 = np.asarray(duplicate_dates[1])
value_means = []
for i in range(len(converted)):
    value_mean = np.mean(values[i])
    value_means.append(value_mean)
value_means = np.asarray(value_means)
```

In [138]:
```
converted= np.asarray(converted)
plt.scatter(converted, value_means)
plt.plot(np.unique(converted), np.poly1d(np.polyfit(converted, value_means, 1))(np.unique(converted)))
print(np.poly1d(np.polyfit(converted, value_means, 1)))
```

```
-0.01236 x + 2375
```



Cell [135] and [137] organizes the data to be plotted. This part of the analysis takes the mean of each array, representing all of the data of the export region at a given time as one value. It also sets up the converted dates to be used as the independent variable, time.
Cell [138] scatter plots the converted date values and average NDVI values at a given time. Then it creates a line of best fit through the points. In this case, the vegetation index decreases by 0.01236 each day in that particular region. Currently, we are utilizing a line of best fit, but a machine learning regression will be implemented to more accurately model and predict future trends.

EVALUATION

Value Proposition:

In the current market, algorithms are not usually patented and sold for money-- algorithms are usually open-source code that only require citations. TerraTracer will most likely follow the same value path as other algorithms. However, TerraTracer, with its increased efficiency could also potentially be patented by the USPTO, which would bring in revenue that would be distributed to deforestation awareness charities. This patent could prove instrumental in the valuation of TerraTracer and its capability to raise money for deforestation efforts.

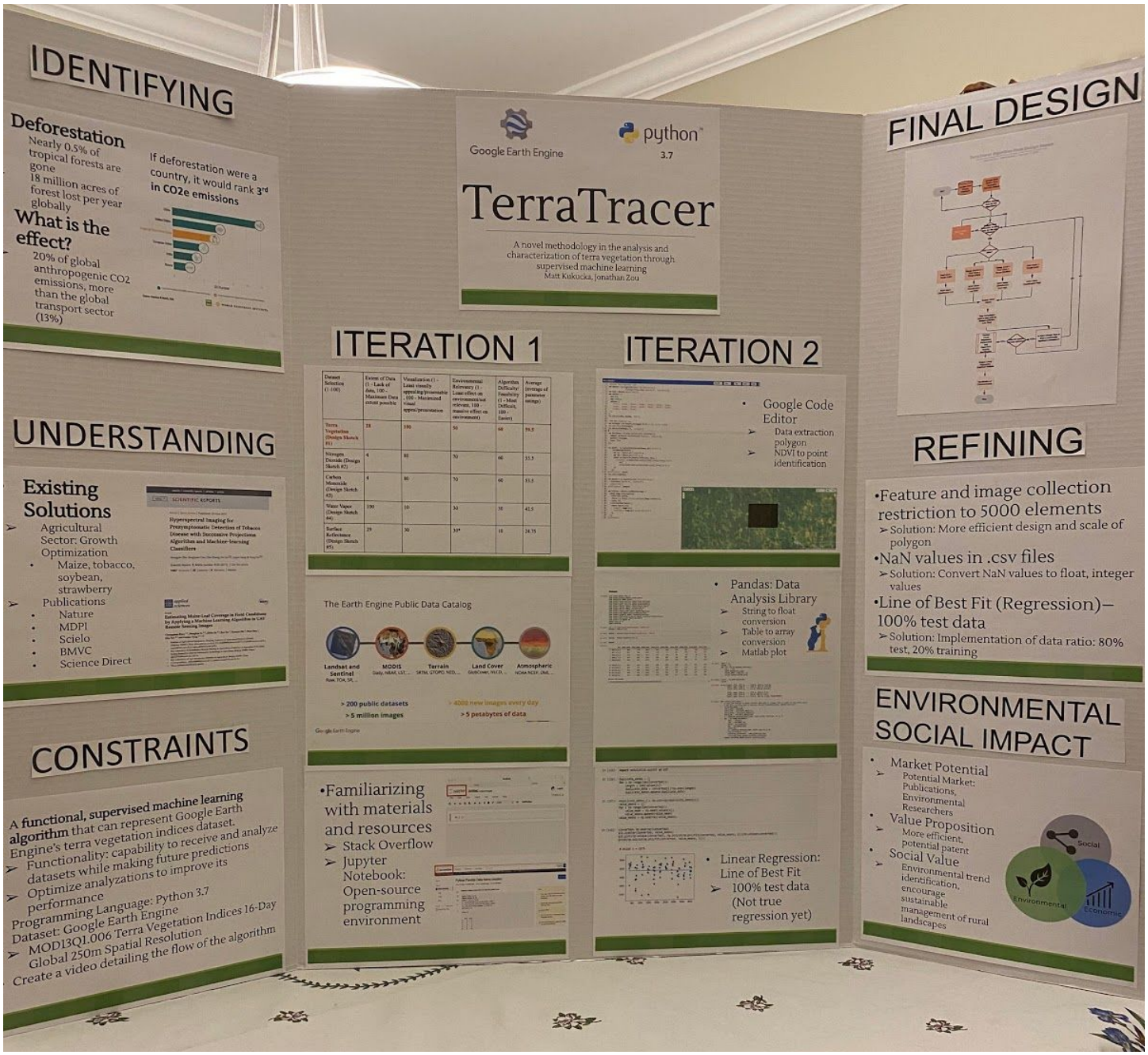
Market Potential:

This invention can prove crucial to publications and environmental researchers, who can be incentivized by the efficiency of our potentially patented algorithm. Algorithms such as TerraTracer are cited by hundreds of researchers and universities for research and awareness programs, which could promote green activism all around the world. The algorithm can also be used pragmatically and not just for research purposes: tracking leaf coverage in certain areas can encourage targeted green activism and action based on the data and trend identified by the algorithm.

Social Value:

TerraTracer is a crucial algorithm in environmental trend identification, especially when it comes to terra vegetation. With its imaging and processing efficiency, TerraTracer could encourage sustainable management of rural landscapes, especially in areas that rely upon sustenance farming. The algorithm has a broad range of social impacts such as integration of sustainable management systems, which could dramatically reduce deforestation as well as helping those in the communities the system is implemented in; these systems have a direct effect on the environment since they slow down sequestered carbon release from trees and improve the agricultural landscape of more rural areas that are looking to clear trees for farmland.

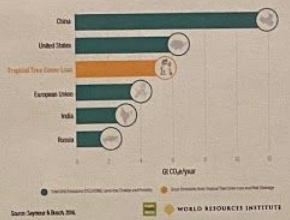
Trifold:


IDENTIFYING
Deforestation
Nearly 0.5% of tropical forests are gone
18 million acres of forest lost per year globally
What is the effect?
20% of global anthropogenic CO2 emissions, more than the global transport sector (13%)
If deforestation were a country, it would rank 3rd in CO2e emissions
Google Earth Engine
python 3.7
TerraTracer
A novel methodology in the analysis and characterization of terra vegetation through supervised machine learning
Matt Kukucka, Jonathan Zou
FINAL DESIGN
UNDERSTANDING
Existing Solutions
Agricultural Sector: Growth Optimization
Maize, tobacco, soybean, strawberry
Publications
Nature
MDPI
Scielo
BMVC
Science Direct
ITERATION 1
The Earth Engine Public Data Catalog
Landsat and Sentinel
MODIS
Terrain
Land Cover
Atmospheric
> 200 public datasets
> 5 million images
> 4000 new images every day
> 5 petabytes of data
•Familiarizing with materials and resources
Stack Overflow
Jupyter Notebook: Open-source programming environment
ITERATION 2
Google Code Editor
Data extraction polygon
NDVI to point identification
Pandas: Data Analysis Library
String to float conversion
Table to array conversion
Matlab plot
Linear Regression: Line of Best Fit
100% test data (Not true regression yet)
REFINING
•Feature and image collection restriction to 5000 elements
Solution: More efficient design and scale of polygon
•NaN values in .csv files
Solution: Convert NaN values to float, integer values
•Line of Best Fit (Regression)– 100% test data
Solution: Implementation of data ratio: 80% test, 20% training
CONSTRAINTS
A functional, supervised machine learning algorithm that can represent Google Earth Engine's terra vegetation indices dataset.
Functionality: capability to receive and analyze datasets while making future predictions
Optimize analyzations to improve its performance
Programming Language: Python 3.7
Dataset: Google Earth Engine
MOD13Q1.006 Terra Vegetation Indices 16-Day Global 250m Spatial Resolution
Create a video detailing the flow of the algorithm
ENVIRONMENTAL SOCIAL IMPACT
Market Potential
Potential Market: Publications, Environmental Researchers
Value Proposition
More efficient, potential parent
Social Value
Environmental trend identification, encourage sustainable management of rural landscapes
Social
Environmental
Economic

IDENTIFYING

- **Deforestation**
  - Nearly 0.5% of tropical forests are gone
  - 18 million acres of forest lost per year globally
- **What is the effect?**
  - 20% of global anthropogenic CO_2 emissions, more than the global transport sector (13%)

If deforestation were a country, it would rank **3rd in CO2e emissions**

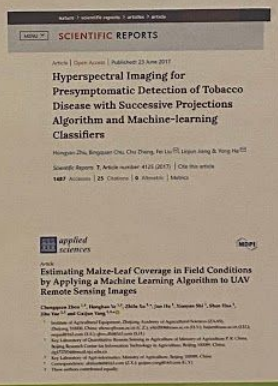


ITERAT

Dataset Selection (1-100)	Extent of Data (1 - Lack of data, 100 - Maximum Data extent possible	Visualization (1 - Least visually appealing/presentable, 100 - Maximized visual appeal/presentation	
Terra Vegetation (Design Sketch #1)	**28**	**100**	**5**
Nitrogen Dioxide (Design Sketch #2)	4	80	70
Carbon Monoxide (Design Sketch #3)	4	80	70
Water Vapor (Design Sketch #4)	100	10	30
Surface Reflectance (Design Sketch #5)	29	30	30*

UNDERSTANDING

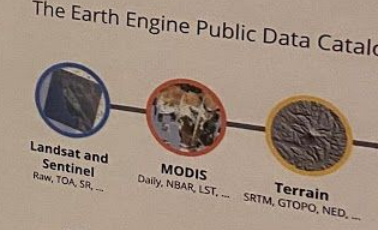
- **Existing Solutions**
  - Agricultural Sector: Growth Optimization
    - Maize, tobacco, soybean, strawberry
  - Publications
    - Nature
    - MDPI
    - Scielo
    - BMVC
    - Science Direct


SCIENTIFIC REPORTS

Hyperspectral Imaging for Presymptomatic Detection of Tobacco Disease with Successive Projections Algorithm and Machine-learning Classifiers

applied sciences

Estimating Maize-Leaf Coverage in Field Conditions by Applying a Machine Learning Algorithm to UAV Remote Sensing Images



CONSTRAINTS

- A **functional, supervised machine learning algorithm** that can represent Google Earth Engine's terra vegetation indices dataset.
  - Functionality: capability to receive and analyze datasets while making future predictions
  - Optimize analyzations to improve its performance
- Programming Language: Python 3.7
- Dataset: Google Earth Engine
  - MOD13Q1.006 Terra Vegetation Indices 16-Day Global 250m Spatial Resolution
- Create a video detailing the flow of the algorithm

- Familiarizing with materials and resources
  - Stack Overflow
  - Jupyter Notebook: Open-source programming environment

Google Earth Engine

python 3.7

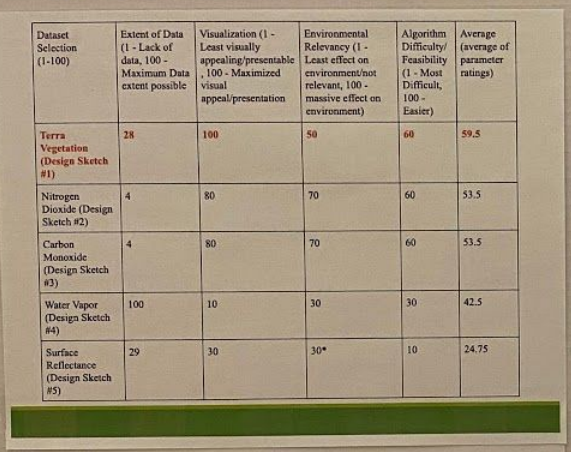
TerraTracer

A novel methodology in the analysis and characterization of terra vegetation through supervised machine learning

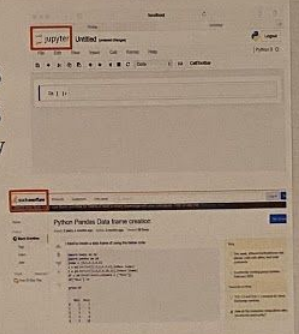
Matt Kukucka, Jonathan Zou

ITERATION 1

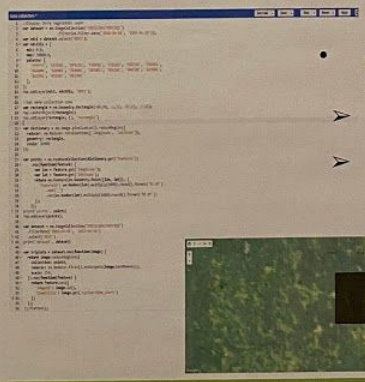
Dataset Selection (1-100)	Extent of Data (1 - Lack of data, 100 - Maximum Data extent possible	Visualization (1 - Least visually appealing/presentable , 100 - Maximized visual appeal/presentation	Environmental Relevancy (1 - Least effect on environment/not relevant, 100 - massive effect on environment)	Algorithm Difficulty/ Feasibility (1 - Most Difficult, 100 - Easier)	Average (average of parameter ratings)
Terra Vegetation (Design Sketch #1)	**28**	**100**	**50**	**60**	**59.5**
Nitrogen Dioxide (Design Sketch #2)	4	80	70	60	53.5
Carbon Monoxide (Design Sketch #3)	4	80	70	60	53.5
Water Vapor (Design Sketch #4)	100	10	30	30	42.5
Surface Reflectance (Design Sketch #5)	29	30	30*	10	24.75



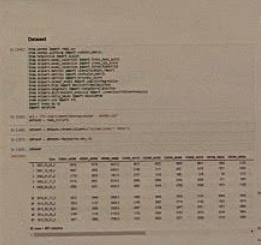
- Familiarizing with materials and resources
  - ➢ Stack Overflow
  - ➢ Jupyter Notebook: Open-source programming environment



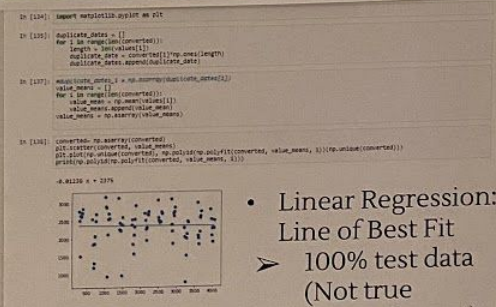
ITERATION 2



- Google Code Editor
  - ➢ Data extraction polygon
  - ➢ NDVI to point identification



- Pandas: Data Analysis Library
  - ➢ String to float conversion
  - ➢ Table to array conversion
  - ➢ Matlab plot

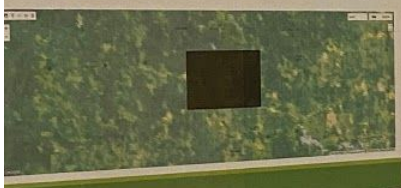


- Linear Regression: Line of Best Fit
  - ➢ 100% test data (Not true regression yet)

r

RATION 2

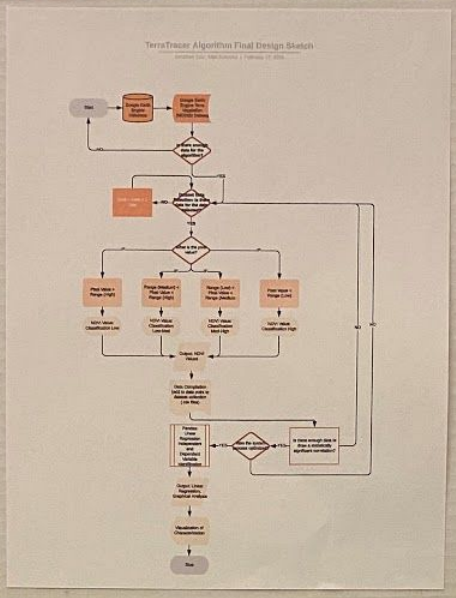
- Google Code Editor
  - ➢ Data extraction polygon
  - ➢ NDVI to point identification



- Pandas: Data Analysis Library
  - ➢ String to float conversion
  - ➢ Table to array conversion
  - ➢ Matlab plot

- *Linear Regression: Line of Best Fit*
  - ➢ *100% test data (Not true ression yet)*

FINAL DESIGN



REFINING

- Feature and image collection restriction to 5000 elements
  - ➢ Solution: More efficient design and scale of polygon
- NaN values in .csv files
  - ➢ Solution: Convert NaN values to float, integer values
- Line of Best Fit (Regression)– 100% test data
  - ➢ Solution: Implementation of data ratio: 80% test, 20% training

ENVIRONMENTAL SOCIAL IMPACT

- Market Potential
  - ➢ Potential Market: Publications, Environmental Researchers
- Value Proposition
  - ➢ More efficient, potential patent
- Social Value
  - ➢ Environmental trend identification, encourage sustainable management of rural landscapes