Geospatial Monitoring of Cropland Health for a Changing Climate

Project Authors (Team SaFe MeMe):

- Sara Hunsberger (shunsbe2@jh.edu)
- Fernanda Montoya (fmontoy1@jh.edu)
- Meucci (Deerspring) Ilunga (milunga2@jh.edu)
- Meklit Yimenu (myimenu1@jh.edu)

Key Research Question: How have variations in key climatic variables—such as temperature, precipitation, humidity, solar radiation, and cloud cover—impacted the health and productivity of croplands across different regions of the United States over time? And what potential does this sort of geospatial statistical modeling show for identifying regions with future risks for poor crop performance?

The objective of our project is to provide a basic analysis of how climate change has impacted regional cropland health and productivity in the United States over the last 25 years. This investigation is motivated by the significant role that agricultural output plays in food security and the broader economy. Understanding climate-driven changes in cropland productivity is essential for all community stakeholders, including policymakers and farmers, to make informed decisions and adapt effectively. To achieve these objectives, our project aims to integrate three key datasets: one for mapping US croplands, one serving as a proxy for vegetation health through metrics such as the Normalized Difference Vegetation Index (NDVI), and one for climate data, including temperature and precipitation.

Original Goals and Project Accomplishments

Our original project proposal goals included acquiring the geographic and climate data needed to assess changes to cropland over time, performing exploratory analyses of the data, and using the data to predict future trends in cropland. At the finalization of our project, we were able to obtain the data required for one state, Illinois, and perform exploratory analyses and predictive modeling. Specifically, Illinois data was gathered and combined from earth engine and was further reduced for exploratory data analysis. Collecting the data from earth engine posed many difficulties because of the size of the data and so multiple programming paradigms were used in this section. This is discussed in further detail in the sections “Overview of Technical Challenges” and “Programming Paradigm Integration”.

After the data was collected we performed data processing and an exploratory data analysis on the data. Data preprocessing and quality checks were conducted on the raw datasets spanning the years 2000 to 2023. Due to their size, the preprocessing was performed in two groups for efficiency. The data underwent assessment to ensure quality, including checking for null values, duplicate rows, range validity, and consistency in units and data types. Outliers were identified and addressed as necessary. Missing values were imputed using group means, after which the datasets were reduced by calculating daily means. This reduction was performed before combining the processed datasets into a single, comprehensive dataset for downstream analysis.

Exploratory Data Analysis focused on understanding relationships between variables in the dataset, particularly the relationship between NDVI (the outcome of interest) and other features. Initial visualizations revealed significant patterns and trends. For instance, temperature appeared to be positively correlated with NDVI while the relationship between NDVI and precipitation was less clear.The analysis also highlighted high levels of correlation among several variables, underscoring the need for careful feature selection to avoid redundancy and multicollinearity.Feature selection began with the computation of correlation coefficients to identify and remove highly correlated predictors. K-means clustering was then applied to the scaled data to group observations and identify features most critical to the outcome, enhancing the interpretability and utility of the final model.

The cleaned and condensed dataset was then used for the construction of predictive model to forecast trends in cropland health. We performed predictive modeling with the use of time series linear models that were fit with the R package fable. Fable allows for time series data to be read into a linear model; time series data is formatted with the use of the tsibble R package that constructs the dataset into recognizable time series components for the fable model to use.

Using the time series linear model, we constructed five predictive models, one for each cropland health of interest (NDVI, EVI, ET, FPAR, and LAI). The model used a time trend and climate variables of interest as predictor variables. We split the dataset into a training and test set (roughly 80% and 20% respectively), where the first 20 years of data was used to construct the predictive model, and the testing set was used to assess predictions for the final 3 years of data. The model performance for each outcome of interest was assessed by comparing the model performance to the true data values. Overall, our models were able to reasonably predict the trends of the testing set.

Existing Work

There has been research done on the impact of climate change on vegetation in the United States. For instance: the article "Impacts of global change on peak vegetation growth and its timing in terrestrial ecosystems of the continental US" by Ying Liu et al. describes an analysis using similar factors that we will be looking at. These factors include how the Normalized Difference Vegetation Index (NDVI) changes with differences in temperature, precipitation and cloud cover. However, this research was done only on *all* vegetation in the United States, whereas our group is more interested in limitng our analsysi scope to the climate impact on croplands specifically. The Liu et al. study found that precipitation had the largest impact on NDVI, which tells us that geospatial precipitation data maybe be a prime data metric to look at for our analysis. [@LIU2021103657]

In terms of looking at climate change and cropland specifically, the paper "Climate change and adaptation in agriculture: Evidence from US cropping patterns" by Xiaomeng Cui talks about climate change's effect on the types of crops being planted; for example in the US, soybean and corn production has increased, and this paper says that climate change has contributed to that increase. This means that in our analysis we can also possibly examine shifts in crops along with changes to their NDVI and other metrics of crop health. [@CUI2020102306]

Overview of Technical Challenges

The biggest technical challenge we faced in doing this project was the size of the dataset we decided to work with. We had originally planned on doing a global or CONUS crop analysis, but when we started to account for spatiotemporal resolution, we realized the dataset we would pull from the Google Earth Engine API be dozens of terabytes in size. While it was in principle possible to work with this full data set, we realized limitations with RAM on our local PCs would simply make it untenable to do the original analyses we had proposed with the degree of flexibility we cared about. We briefly considered basic multiple linear regression using the biglm R package as one workaround for one key analysis, but ultimately decided to pivot to a more limited (but flexible) scope. Thus, solving this technical challenge involved reducing the region of geographic interest to just the state of Illinois, reducing spatiotemporal data resolution to 1km x 1km land area on a weekly basis, and using the arrow package to store the final dataset. The final dataset size was 12gb, which could more easily be loaded into local memory on our devices, and our methodology scales in just such a way that the pipeline could be reused for different geographic regions if ever desired: a reasonable comprise over our original proposal.

Programming Paradigm Integration

To meet project requirements, we decided to use three programming paradigms to implement our proposal. Functional Programming was a requirement inherent to working with the GEE API as the API uses an inherently functional style. That is: processing the GIS data we wanted via the API required writing local image preprocessing functions (lambdas, or otherwise) applicable to thousands of image instances that remained server-side. As showcased in our final presentation, this resulted in writing several local functions that were handed over to the server by the API for running on multiple data objects through a single *.map() call, which would apply what ever preprocessing function we wrote to ~1000 image data objects simultaneously. These functional calls via the API collected and pre-processed thousands of GIS data images simultaneously, which we could then download. However, based on speed of API access, all processing and download operations for the 1244 final image dataset took around 8 hours, which we decided to implement using a remote server. To do this, we relied on a Shell Scripting Paradigm: we wrote a bash script to run on a remote Linux server to make repeated calls to our GEE API script which was implemented using the functional programming approach described above. This allowed us to collect our dataset over an 8 hour period, via running a short bash script that could then be run in the background. This was useful as our team could not find a good way to otherwise keep one of our computers running for long enough to get all the data processed and downloaded. However, when the data was finally collected from the server, the final preprocessing step required standardizing the form of the dataset; to do this, we leaned on the Object Oriented Programming paradigm and created a custom python object to load in the raw GIS data images pulled from the API, then implemented several methods for the custom GIS object class we created designed to flatten the GIS image data into an R-compatible data frame, which we then stored as a compressed Arrow file in R. Loading the data into a single unified object in memory with applicable methods defined for the class was the best and most natural approach we saw to store, analyze, and convert the raw API GIS data into an R-compatible data format.

Data Product

We made a website using quarto that describes the data that we used, our steps for data collection and cleaning, an exploratory analysis of the data and finally how we created our final predictive model. The link to the website is below and all of the code for the website is in the Website folder of this repository. All other code for the project is contained in this repository, with an explanation of each folder in the ReadMe file.

Project Links

Website: https://sarahunsberger1.github.io/Project4_website/ (source)

Github Repository: https://github.com/jhu-statprogramming-fall-2024/project4-safe-meme (source)