# Predicting and Visualizing United States Forest Growth

**Adam Kirsh**
akirsh@seas.upenn.edu
**Nate Rush**
narush@seas.upenn.edu
**John Mason Mings**
jmings@seas.upenn.edu
**Maria Turner**
maturner@seas.upenn.edu
**Ethan Perelmuter**
peethan@seas.upenn.edu

## Abstract

Sapling (sapling2020.com) helps everyday people understand how the environment in the United States is changing by visualizing forest growth and loss. Sapling generates and displays a heatmap representing changes in forest coverage between 2016 and 2019 in any region within the continental US that is selected by users. This forest growth and death heatmap is generated from a decision tree model trained on satellite images and forest data from a 2008 USGS forest survey. To our knowledge, this is the first forest growth-death visualization capable of near-real-time heatmaps automatically from open-source satellite data. Sapling's intention is to make forest growth/death more concrete, and in turn inspire environmental activism. We hope this tool will help show the ways in which computer science techniques can augment existing climate data and address anthropogenic climate change.

## 1 Motivation and Functionality

Many people in the United States are aware of the immediate need to take climate action, however, this does not translate into the desired level of action. Sapling's goal is to make the effects of climate change more concrete and thus more likely to lead to action. Concreteness is "how specific, definite, and vivid" something is, and concrete communication has been shown more likely to lead to action. Once a user internalizes what is happening to the environment, Sapling allows them to donate to highly rated environmental charities.

The primary product functionality is in viewing a map of the continental United States and choosing an area to analyze or choosing from preset locations. The map shows the change in forest coverage over time represented as colors on a spectrum from red to neutral to green. The darkest red represents the greatest forest loss while the brightest green represents the greatest forest growth. Users can click buttons to view the effects of individual events like a particular wildfire in California. Clicking the "California Wildfires" button transports the map to one of the wildfire sites. From this view, the change in forest coverage and the effects of the wildfire are apparent. For these presets, information like government and advocacy groups' responses, how the current situation can be helped, and how the likelihood of these events can be decreased long-term and recommended charities for these areas.

## 2 Related Work

Across all case studies, input data typically consisted of LiDAR, satellite, spectral, and Landsat data, indicating significant data preprocessing steps. Preprocessing included variable extraction and selection, pixel manipulation, featurization, and rasterization of LiDAR data. Models trained were typically 2D or 3D convolutional neural networks (CNNs) used in tandem with SVM classifiers, decision trees, and different regressions. Several papers provided useful comparisons of model performances in predicting metrics similar to forest coverage, helping inform our model choices.

Applications of this form of analysis relating to our goal of promoting awareness for climate change included monitoring habitats, detecting oil spills, characterizing and generating forest inventories, classifying terrain, and forming more precise methodologies for meeting sustainable development goals. For example, one paper developed a model to detect oil spills using LiDAR data and satellite data more accurately than using

conventional detection techniques. By developing models to do analysis using several types of input data, researchers have been able to predict environment-related metrics with higher accuracy, and have been able to extend these predictions to areas where current satellite data and/or LiDAR data is not available. These papers validated the need for a model like ours, as we seek to detect forest coverage in the absence of recent satellite data.

An informative paper on using satellite imagery to predict poverty further solidified our understanding of how we might use a CNN or regression to best predict metrics from satellite data (Jean). This paper trained a CNN to identify nighttime image features that indicate variation in economic outcomes, then trained a ridge-regression model to synthesize the results of the CNN with daytime data. The scarcity of training data led to the necessity of a transfer learning approach. We found this paper informative in outlining the possible risks we could face sourcing carbon stock training data.

Another highly relevant paper on synthesizing disparate LiDAR and satellite datasets through deep learning provided a lot of context for how we might approach our model accuracy goals and highlighted the need for a model to predict forest coverage which does not require input LiDAR data (Ayrey). The paper detailed how LiDAR-derived forest inventories are uncommon at a regional scale and also less accurate than a modeling approach using a three-dimensional CNN. This approach was most successful in estimating biomass and densities. While this paper demonstrated the effectiveness of using a model to estimate biomass instead of using manually collected data, it failed to account for the cases in which there is no LiDAR data available. Thus, this approach is inadequate for our task, as we seek to monitor forest coverage across all areas, including economically developing countries which may not LiDAR infrastructure in place.

A full list of relevant academic papers consulted can be found in the references.

## 3 Technical Approach

Our technical approach can be broken down into three main components: the tree growth/death model, the Linux server, and the cache on the server. We will explore the technical details of each component, as well as how they interact with each other.

### 3.1 Tree Growth and Death Model

The tree growth/death model is the main technical component in our project. Abstractly, it can be thought of as a function that takes a (latitude, longitude) pair as input, and returns a JSON list of objects with the fields (latitude, longitude, growth/death). Concretely, these JSON objects represent a growth or death of forest material occurring at the given latitude and longitude coordinates. The model was entirely implemented in Python, due to useful satellite data processing libraries as well as comparative efficiency to JavaScript.

### 3.2 The Satellite Data Pipeline

Training this model, as well as running this model, requires satellite data. After exploring a variety of alternatives like the Google Maps API, we settled on the EarthExplorer API as our source of satellite data.

The first major piece of our model is Python code that interacts with the EarthExplorer API. Concretely, this code takes a given latitude and longitude pair, makes requests to the EarthExplorer API to download large satellite raster images for this database in a ZIP file.

Within this downloaded ZIP file, we only need some of the stored files; to reduce the size of stored data (as these files could be as large as 1GB), we extracted the metadata file we needed as well as raster bands 4, 5, and 6. These raster images represent the blue, green, and red color bands specifically, and in turn allowed the next step of the pipeline to convert these raster images to a JPEG image by overlaying the bands. This resulting colored JPEG is both easier to process and more compact to store.

This pipeline was a major source of implementation complexity. The most common and accessible coordinate system is latitude and longitude, and as we will explore below, this coordinate system needed to be both in the input and output

to the model. As such, we needed to convert between latitude/longitude and the other projection systems used internally by EarthExplorer and the satellite images. More specifically, we had to convert to Albert Equal-area Conic projection, then to the Universal Transverse Mercator, and finally back to latitude/longitude. Understanding which conversions where necessary was a major source of implementation difficulty, both from an understanding and evaluation perspective. To accomplish the above conversions, we initially implemented our own conversion code, but after accuracy and rounding errors, we moved to using a Python library.

### 3.3 Training the Model

To train this model, we used a USDA Forest Service data set from 2008 that contains forest/non-forest data across 300m square regions of the United States. We used the aforementioned satellite data pipeline to download randomly selected 2008 satellite data from across this area, and stratified these downloaded JPEG images into testing and training data sets.

With the training data set, we used existing Python data-science libraries to train a variety of simple classifiers. Notably, before the images are fed into these classifiers, they are preprocessed; we lay a grid of 300m squares on the satellite images so that they correspond to the given squares in the forest/non-forest dataset. Then, we take an average of pixel RGB data across this square in the JPEG image. Finally, we feed tuples of these average RGB values as well as the forest/non-forest boolean into both a decision tree and multivariate linear regression model, the two classifiers we tested.

### 3.4 Generating the Prediction Map

Given the satellite data pipeline and the trained model, we are now capable of creating a prediction map for a given latitude and longitude coordinate pair. The prediction map can be understood as a grid on top of the map, where any grid square is true if the area underneath is forest, and false if the grid is not forest.

To create this map for a given (lat, long) pair, first the satellite data pipeline downloads and creates an JPEG image that contains these coordinates. Then, the model is chunked in the same process described in the training of the model above, before these chunks are fed into the trained model. The output of this predictive step (e.g. forest/non-forest) is then stored within the grid, effectively creating the prediction map.

### 3.5 The Change Map

As our model is primarily interested in calculating the growth/death of forests, we must calculate two prediction maps within the same region to see if forests have grown or shrunk. In our case, due to the limitations of the EarthExplorer API, we choose to download images from 2016 and 2019.

Thus, for a given latitude and longitude pair, an image from 2016 and 2019 were downloaded by the data pipeline. Then, two prediction maps were generated - one for each year. Furthermore, as the data pipeline downloaded satellite images from the same location in these two years, these two prediction maps could be directly compared. This comparison can be considered a matrix subtraction, where the 2016 prediction map is subtracted from the 2019 prediction map; if the resulting value is 0 in the square, there was no change. If it is negative, then there was forest loss, and if it is positive, then there is forest growth.

Within this step, there were again significant implementation complexities that came with coordinate conversions, as were explored in the satellite data pipeline, and similar solutions were used.

### 3.6 The JSON Changes List

The final step of the model is to use the generated change map to create an easily-consumable list of JSON objects that summarize these changes.

To do so, the change map was further chunked into pieces so that multiple elements in the map were grouped, which was done for efficiency reasons. Then, if there were above some threshold of forest growth squares in these chunks, a JSON object was output with a growth marked at the latitude and longitude of this chunk. The same was done for forest loss.

This threshold was chosen by manual tuning, again based on ground-truth data sets. More specifically, we used known areas of forest growth

and death, like California wildfires, to tune the threshold to ignore noise and mark only valid loss and gain.

### 3.7 The Server

To display the output of the model to users in a convenient way, we implemented a web-server and front-end for interacting with the model as well as displaying a variety of case studies and metrics.

### 3.8 The Website

The frontend of the website was written in standard HTML, along with JavaScript for interactivity and JQuery code for interactivity. Furthermore, to provide a nice interface to users to interact with the model, we used Google Maps API to create a map page where users could select points on a map of the United States. With the click of a button, they could make a request to the server to display a heatmap of the forest growth/death over the surrounding region.

### 3.9 Serving the Website

The static pages of the website are served from a Express Node server running on a Linux box on the Google Cloud. This linux machine runs a reverse proxy NGINX server to receive requests and forward them to a running localhost server.

The DNS domain records for sapling2020.com, where this website is hosted, is managed using Google Domains.

### 3.10 Running the Model on the Server

One of the routes on the server is the /getRegion route, which takes a latitude/longitude pair as input, and which can be understood as the route that actually runs the model on the server.

When this route is called, the server creates a new process that begins running the Python model. Standard Node utilities are used for creating this process. This model process writes the output list of JSON changes to a file, and then uses standard out to write back the name of the newly generated file. This file is read in by the Node server, which then reads the JSON data from the file and writes it back to the user.

While this model is running, given its runtime, a loading screen is displayed to the user. Furthermore, request timeouts are increased dramatically on this route, so that the long-running request will terminate before the user's request times out. These timeouts were a major, unexpected source of implementation complexity. The reverse proxy server had a second, shorter timeout implemented as a default and was causing errors in some long-running cases, leading to a very hard to debug issue.

### 3.11 The Cache

As is mentioned above, the runtime of the model is very long. As such, a cache was used to avoid running the model when it was not necessary.

Due to the high-precision of the coordinates requested by the user, it was not suitable to simple cache based on coordinates. Instead, we cached data based on the given UTM region that the coordinates fell in, one of the projects the model must parse internally. This cache was implemented as a simple file-store on disk, and as is explored in the evaluation metrics section significantly improved performance after the first request.

## 4 Evaluation

Given the recent COVID-19 pandemic, user interviews and interactions became much harder to engage in - making product evaluation difficult from a qualitative perspective. However, we thoroughly evaluated our product from a technical perspective, with the goal of showing that our product would be able to function as a released, production piece of software generally. The metrics we evaluated our project on fall under two areas of consideration. First, we studied the performance of our model from an accuracy/precision perspective, as well as a run-time perspective. Second, we analyzed the performance of our complete system under a variety of real-world workload conditions with multiple users, multiple concurrent requests to the model and the cache, etc. We will explain, in detail, our evaluation steps and results for each of these below.

### 4.1 Model Metrics

One main technical contribution is our model that can predict if a given 250m region of area has tree cover or not given satellite data as input. We evaluated this model on a variety of metrics.

First, and most importantly, we evaluated how accurate the model is. We wanted to know

how often the model is correct when it outputs tree/no-tree results. To test this, we stratified our ground-truth dataset into a testing and a training dataset. We then trained the model on the training portion, and assessed its accuracy on the testing portion.

We tested the model on 5 randomly selected regions in the western and eastern U.S, with the model performing noticeably better in the Western U.S. than the Eastern U.S. We suspect this is due to biases in our training datasets, as forest patterns differ according to geography and our training data may have been biased towards forest patterns typically found in the Western U.S.

Our goal was to achieve accuracy greater than 75%, and we were able to achieve this in our decision tree model. As shown in Appendix A Figure 1, we achieved an accuracy of 85% using our decision tree model for the Western U.S. and achieved an accuracy of 75% using our multivariate logistic regression model. We had slightly lower accuracies for the Eastern U.S., again demonstrating the biases in our training data. An area for future improvement here would be to continue expanding our training dataset with more diverse images to encompass larger portions of the United States and improve the model's accuracy.

Next, we evaluated the running time of the model. Making a prediction through the model requires inputting satellite imagery and getting an output tree/no-tree classification. As this is a major piece of the entire system that users interact with, its running time is crucial. We broke down the running time of each step in the model to better understand where we were experiencing bottlenecks, as suggested by some of our evaluation plan peer feedback, and this helped inform our decision to cache our data.

We used basic Python timing tools to run the model five times in different locations and record the average time that each step took. Within the total average runtime of 21.4 minutes, about 70% of this time is spent within the download step at 15.3 minutes, where the network is the limiting factor. As shown in Appendix A Figure 2, this step is the bottleneck in our running time. The final change map and JSON output conversion steps take up another 20% of the running time but are configurable to run faster at a lower resolution.

Notably, though this model was trained on local computers, it now runs entirely in the cloud on our server. It is thus necessary that this model performs well with respect to the above metrics in this environment.

### 4.2 Full System Metrics

Other than the model, there is a surrounding frontend and backend that allows users to a) visualize a map of tree growth and death and b) make requests to the model to visualize these statistics on specific regions. The second part of our evaluation consisted of evaluating how effectively this system would allow users to access this information.

First, we collected metrics that capture the quantitative experience of a single user using the system. The first metric is the average load-times of the various pages users interact with. We used existing tooling, namely pingdom.com, to measure the load-time of the various pages of our website from around the U.S., and see how this relates to our server location.

As illustrated in Appendix A Figure 3, we found that our landing page had a load time of 158 ms, which is considered excellent, whereas our map page had a load time of 349 ms, which is still within an acceptable range, and comparable to the load time of Google Maps.

Second, we measured the run-time of requests that a user makes to visualize specific areas of tree growth/death. There are notably two routes that a request can take: it may be one of the cached locations, or the model may need to run on new satellite data. This evaluation metric focuses on highlighting the difference between the cached locations and new computation and highlights how much longer one takes than the other for a user.

We incorporated this metric to address some of the peer feedback given for our evaluation plan, as many peers expressed concern that the running time of the model would significantly hamper the user's experience. This evaluation demonstrates how that issue is largely resolved with the use of

caching.

Our results for this portion of the evaluation are shown in Appendix A Figure 4. Without a cache, the running time is the same as the running time of the model at 21.4 minutes, but once the data is added to the cache, the heatmap can be built in about 12 seconds, which is a speedup of about 100x.

Finally, we evaluated how our system performs in the context of multiple users making requests at once. We performed multiple uncached requests to the server at the same time, tested how long each request took, and then averaged the response time. Our goal was to reflect how response times change as more users make requests at once and what the limits are of our system in terms of concurrent users.

As shown in Appendix A Figure 5, the average response time increases by almost 1.5x when there are four concurrent requests instead of just one. Furthermore, four or more requests to the server simultaneously result in increased error rates and timeouts.

### 4.3 User Research and Evaluation

We evaluated our product with ten unique target users by conducting video interviews. We first evaluated how easy they found the product to understand and use, by giving them no instructions and allowing them to use the product in a free-form setting. We then asked the user to record exactly what actions they took and in which order.

We found that users typically spent a large amount of time on the landing page reading through the instructions, which was not what we had initially anticipated would happen. In response, we reconfigured the content and created separate pages for our pre-set case studies.

The user then typically went straight to the heat map and selected the Walker Fire pre-set case study, viewing the cached results and spending an average of 25 seconds interacting with the heat map. Then, the user exited out of the case study and selected a custom region.

We stored the coordinates of these custom regions that were selected, and upon further examination, found that 9/10 users selected areas they were already intimately familiar with. We believe this represents an increased curiosity about forest changes. Users attempted to tie together their lives and this new data, indicating a newfound curiosity, and meeting our established goal of making climate change concrete.

Finally, we had our users fill out a short survey about the areas they discovered, testing basic metrics such as whether they recalled the approximate extent of forest growth and loss shown and whether they retained any information from the case study they read. We found that users were able to recall the proportion of forest growth and loss with an error margin of 15%, as measured by the ratio of growth to loss heat map markers on the map. This shows extremely high retention rates of quantitative information, further demonstrating our success in making climate change more concrete.

Given the opportunity to conduct further user evaluation, we would have increased our sample size and conducted more thorough interviews, but this proved difficult given the changes in circumstances. While we recognize that we were not able to definitively show that users were more likely to donate to our selected charities upon interacting with Sapling, we believe that our evaluation results demonstrate how Sapling both sparks the average user's curiosity and promotes retention of abstract information.

### 4.4 Evaluation Conclusions

For the model accuracy, we were able to achieve our goal of 75% or better predictions of tree/no-tree. For the full system, we had set an initial goal of an average tested load time of all pages on the website to be less than two seconds, a standard maximum allowable load time for many users. We extended this goal load time to all cached computations as well. We were successful in both regards, as shown in the figures included in the appendix.

In the case of live computation through the model, due to the large amount of computation that must occur for the model to process this satellite image, we had set a goal to process this new image

within a reasonable amount of time for the user to not leave the site. We had anticipated this to be within five minutes but were unable to meet that goal due to the bottlenecks in downloading the data.

## 5 Societal Impact

Sapling is a website meant to drive awareness in a time of climate crisis, however, the interconnected nature of technology sometimes leads to unforeseen consequences. Sapling has minimal privacy and data security concerns, yet it has a medium risk of being misused by illegal logging operations. All information that Sapling is trained on and uses within its model is open-sourced and publicly accessible. We have also followed the AWS cloud security best-practices guide, specifically in regards to our server access management.

Any type of personally-identifying information deserves to be treated with extreme care. Breaches of this type of data in today's world are very serious. We have opted to stay away from all features requiring personally-identifying information, and as such Sapling does not store any personally-identifying information for living creatures other than trees. We also do not store any information about user sessions. Although storing "cookies" may enable certain features on Sapling, we deemed this risk unworthy. Not only does this guarantee that we won't spill any of this data, but it reduces the likelihood of other types of attacks. Simply, there's less potential reward for compromising the product.

In regards to server security, to prevent access by malicious parties (e.g. may begin logging user sessions, etc), we have followed the AWS best practices guide for server management. Specifically, only certain SSH keys can access the server, and these keys are stored by team members in their respective password managers as well as only locally on their computers. Finally, we do not accept any user input to be displayed on the website, making various injections as well as cross-site scripting attacks impossible.

The data presented on the map is at a resolution such that only macro-geographic features could be made out. These include locations of cities, bodies of water, or other landforms like mountains. It is not possible to use the map data to determine the location of individual buildings and reveal to the public where isolated structures are.

If a group intended to conduct large-scale illegal logging, they could use Sapling to ensure they stay under-the-radar by monitoring forest coverage in the areas they log. Although unlikely, this would have the opposite climate effect as is intended. This would also leave groups in the immediate area of illegal logging vulnerable to displacement or violence.

Additionally, if our data were compromised, this product could be used to hide changes or show better forest coverage than exists. This could be used to lobby for looser environmental regulations or simply uninspire climate activists. Finally, there is a long-term risk of demotivation, assuming high adoption of Sapling. It is possible that those tracking forest coverage through Sapling might see short term progress and then reduce their efforts and proclaim climate victory. However, reversing or minimizing anthropogenic climate change is a huge task that would require years of "progress" on Sapling. It is not the intent of our product to create a sense of false victory, however, it is possible.

## 6 Discussion

We believe Sapling will continue to be a tool for social good in the future. There are a variety of ways we'd like to continue improving Sapling both technically and with more of a user focus.

Firstly, we'd like to improve our prediction of forest coverage change. We trained multiple models, but believe for the best results the US should be covered by multiple models. Geographic factors provide inherent differences in the satellite image patterns from around the country that would be best accounted for by separate models, rather than inside of one predictive model. This would also allow more easily for the addition of state or region specific datasets for training. Additionally, the most common and most severe piece of user feedback we received pointed to long wait times. The caching system we put in place improved the wait times by about 100x, however it is realistic to preprocess and cache the entire map. This would require a financial investment up front for the computational resources. Preprocessing the entire

map would also require either a larger server for extensive caching or implementation of a content delivery network (CDN).

We also believe that coordinate conversions and translating projection systems deserves more up front attention than originally given. Managing these translations proved one of our greater implementation challenges. For future projects using similar approaches, we recommend creating a standalone coordinate handling module. Users testing our product commented on how concerned they were after use, and team members observed increased curiosity. Particularly, users consistently tried the custom change map feature. They were interested in viewing the forest changes for areas they were intimately familiar with - their hometowns, current living areas, or national parks. We believe this is a positive indicator towards sparking action. In the future, users should be able to further craft their searches by selecting custom timeframes.

Further user engagement should be built by allowing users to favorite areas and receive ongoing email or text updates about them. User engagement can also be encouraged by real-time tracking during traumatic events like large wildfires or hurricanes.

There is also an opportunity to more closely include one stakeholder group. Sapling could partner with tree planting and regrowth initiatives to help decide which areas are most in need of replanting. Some of these organizations include 8 Billion Trees and Team Trees.

# 7 Business Analysis

## 7.1 Market Opportunity

Sapling has gone through many versions including one stark shift due to COVID-19. In the beginning of this semester, we had planned on delivering a product to the National Forest Service and had conversations with other private organizations who showed partnership interest. As the COVID-19 stress grew on these organizations, they apologetically communicated that their priorities were now elsewhere. Sapling then pivoted to its current goal and implementation of making climate change more concrete and inspiring environmental activism. For the remainder of this report, the old business model will be touched on while the current implementation receives more attention.

The original Sapling business model relied on a lack of automated environmental analysis for interested parties. Many organizations like the National Forest Service, private growth yards, land insurers, and others make key decisions based on land changes. In many cases, these decisions are long standing and have high financial impact. For instance, when the National Forest Service redistricts land it is set for 30 years. As such, these parties are highly motivated to acquire relevant data.

Currently, many of these organizations collect data by hand. This means sending workers out into parks and private forests to survey the land. This process is time and resource intensive. This data can also become outdated quickly in the case of significant events like wildfires or become outdated more naturally in a few years. The motivation for the newer and present version of Sapling has been discussed extensively. Despite 61% of Americans claiming to be concerned about climate change, few act. This is the unfortunate opportunity Sapling now addresses.

## 7.2 Users and Customers

Today's Sapling differentiates between customers and users. Users are the everyday people we hope to inspire towards environmental activism. Given the nature of our tool, we anticipate the largest user demographic to be young, tech forward, professionals and students. While our goal is to change the perception of climate change as an academic pursuit, we believe our tool still has a somewhat techy orientation that will filter out older age groups and some less tech-savvy young people. Future versions of the product should address this.

The organizations funding us will be environmental grant writers like The Green Climate Fund, Green Grants, and The Climate Works Foundation whose missions fall in line with ours: help our odds against climate change. Discussions with them in the early fall confirm that they agree with our current approach. Since we are helping implement their mission, they are our customers.

### 7.3 Market Segment

It is difficult to estimate the total of environmental grants afforded in the United States each year since many are handed from private organization to private organization, however Yale Research indicates a minimum of $218.5 MM was spent lobbying congress in favor of the environment between 2000 and 2016. Open Secrets, an organization dedicated to political transparency, notes that "[a]s attention to climate and resource issues has increased in recent years, environmentalists have grown far more influential in Washington, even if political contributions from environmental groups are but a fraction of those given by the industries they generally oppose". This market direction favors Sapling.

### 7.4 Competition

Aside from research studies conducted within universities, key competitors include paid services which do similar forms of analysis, tracking forest metrics and reporting real-time data to clients. Examples of these include Forest Business Analytics, Remsoft, and Forest Economic Advisors.

Forest Business Analytics (FBA) is a consulting service that has a stated goal of "help[ing] customers to better understand their present operating business environment through diligent analysis and application of tailored solutions." The company's target market is forest-related businesses and the main form of analysis it conducts is economic in nature.

Similarly, Remsoft is a paid service that caters to the forestry industry, offering operational consulting services and "power[ing] critical decisions for forest-to-mill planning, land management, and MRO inventory optimization". While the company conducts similar forms of analysis such as modeling wood flow and other forest metrics, it performs this analysis for economic purposes such as determining the value of timberland investment properties.

Forest Economic Advisors is a consulting service that conducts the aforementioned forms of analysis with a heavy emphasis on "economic forecasts, lumber, timber, panels, and other wood products."

These companies perform very similar types of forest analysis using satellite data and display their metrics in a palatable interface. However, they do not cater to the same target market segment, instead offering paid services to clients in the forestry industry. They also include an extra layer of analysis on the initial predicted metrics, offering economic interpretations to cater to their clients' needs.

Sapling distinguishes itself in that our target user is the average individual and our target market is environmental grant writers whose missions are to help our odds against climate change. Sapling does not perform economic analysis of forest metrics and does not perform the supply-chain analysis characteristic of its competitors. We do not aim to promote forestry-related businesses, rather conveying the base forest metrics to any individual with some analysis describing the implications of our findings in the grand scheme of the Climate Crisis.

### 7.5 Cost Analysis

Currently, Sapling2020.com runs on a single Google Cloud n1-standard-4 E2 instance. This costs a fixed $97 a month, although these costs are subject to slight variation due to Google Cloud variable pricing. Furthermore, costs scale linearly with used network data, which in turns scales with usage, though this is in the thousandths of a cent per user.

Given our current caching mechanisms, more usage will result in lower network usage of our product, as more requests can be served from the cache and will not need to be generated from downloaded satellite images, the largest network requests performed by our servers. Furthermore, much of our content can be served from a CDN, further limiting costs through caching, and as such, we do not expect monthly expenditures to be more than $150, no matter the scale.

If we were to cache the entire continental US up-front to improve the user experience, this could be done by less than 10 higher-tier/spec servers running on Google Cloud overnight; this would likely cost in the hundreds of dollars, but would be a one-time fixed cost.

### 7.6 Revenue Model

The former version of Sapling would have worked as a product-based consultancy. With our model and basic analytics already built, we had planned on extending the software package with some custom metrics and possible integrations with clients' existing monitoring systems.. These increased variable costs are justified by high average contract size for the old business model.

Today's Sapling will seek grants from organizations like The Green Climate Fund, Green Grants, and The Climate Works Foundation. Climate Works approved $100,000 "to protect primary forests in Papua through monitoring and advocacy efforts" and $115,000 "to support governance and technical requirements for effective carbon pricing in the aviation sector" in 2018. Based on these organizations' portfolios, they would be interested in supporting our project on an annual or 5 year basis.

## 8 Acknowledgements

We thank Dr. Eric Eaton, Dr. David Rolnick, Dr. Jane Dmochowski, Dr. Brett Hemenway Falk, and Dr. Ani Nenkova for their invaluable guidance on this project.

## 9 References

Ahmed, Oumer S., et al. "Characterizing Stand-Level Forest Canopy Cover and Height Using Landsat Time Series, Samples of Airborne LiDAR, and the Random Forest Algorithm." ISPRS Journal of Photogrammetry and Remote Sensing, vol. 101, 2015, pp. 89–101., doi:10.1016/j.isprsjprs.2014.11.007.

Ayrey, Elias, et al. "Synthesizing Disparate LiDAR and Satellite Datasets through Deep Learning to Generate Wall-to-Wall Regional Forest Inventories." 2019, doi:10.1101/580514.

Dalponte, M., et al. "A System for the Estimation of Single-Tree Stem Diameter and Volume Using Multireturn LIDAR Data." IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 7, 2011, pp. 2479–2490., doi:10.1109/tgrs.2011.2107744. Gleason, Colin J., and Jungho Im. "Forest Biomass Estimation from Airborne LiDAR Data Using Machine Learning Approaches." Remote Sensing of Environment, vol. 125, 2012, pp. 80–91., doi:10.1016/j.rse.2012.07.006.

Goetz, Scott J, et al. "Mapping and Monitoring Carbon Stocks with Satellite Observations: a Comparison of Methods." Carbon Balance and Management, vol. 4, no. 1, 2009, doi:10.1186/1750-0680-4-2.

Holloway, Jacinta, and Kerrie Mengersen. "Statistical Machine Learning Methods and Remote Sensing for Sustainable Development Goals: A Review." Remote Sensing, vol. 10, no. 9, 2018, p. 1365., doi:10.3390/rs10091365.

Jean, N., et al. "Combining Satellite Imagery and Machine Learning to Predict Poverty." Science, vol. 353, no. 6301, 2016, pp. 790–794., doi:10.1126/science.aaf7894.

Kim, Yong Hoon, et al. "Machine Learning Approaches to Coastal Water Quality Monitoring Using GOCI Satellite Data." GIScience Remote Sensing, vol. 51, no. 2, 2014, pp. 158–174., doi:10.1080/15481603.2014.900983.

Knudby, Anders, et al. "Predictive Mapping of Reef Fish Species Richness, Diversity and Biomass in Zanzibar Using IKONOS Imagery and Machine-Learning Techniques." Remote Sensing of Environment, vol. 114, no. 6, 2010, pp. 1230–1241., doi:10.1016/j.rse.2010.01.007.

Langner, Andreas, et al. "Integration of Carbon Conservation into Sustainable Forest Management Using High Resolution Satellite Imagery: A Case Study in Sabah, Malaysian Borneo." International Journal of Applied Earth Observation and Geoinformation, vol. 18, 2012, pp. 305–312., doi:10.1016/j.jag.2012.02.006.

Lodha, Suresh K., et al. "Aerial LiDAR Data Classification Using Support Vector Machines (SVM)." Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), 2006, doi:10.1109/3dpvt.2006.23.

Mora, Brice, et al. "Modeling Stand Height, Volume, and Biomass from Very High Spatial Resolution Satellite Imagery and Samples of Airborne LiDAR." Remote Sensing, vol. 5, no. 5,

2013, pp. 2308–2326., doi:10.3390/rs5052308.

Reed, Bradley C., et al. “Measuring Phenological Variability from Satellite Imagery.” Journal of Vegetation Science, vol. 5, no. 5, 1994, pp. 703–714., doi:10.2307/3235884.

Saatchi, S. S., et al. “Benchmark Map of Forest Carbon Stocks in Tropical Regions across Three Continents.” Proceedings of the National Academy of Sciences, vol. 108, no. 24, 2011, pp. 9899–9904., doi:10.1073/pnas.1019576108.

Xu, Zewei, et al. “A 3D Convolutional Neural Network Method for Land Cover Classification Using LiDAR and Multi-Temporal Landsat Imagery.” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 144, 2018, pp. 423–434., doi:10.1016/j.isprsjprs.2018.08.005.

Yuan, Tianle, et al. “Automatically Finding Ship Tracks to Enable Large-Scale Analysis of Aerosol-Cloud Interactions.” Geophysical Research Letters, vol. 46, no. 13, 2019, pp. 7726–7733., doi:10.1029/2019gl083441.

Zhi-Ming, Dong, et al. “Oil-Spills Detection in Net-Sar Radar Images Using Support Vector Machine.” The Open Automation and Control Systems Journal, vol. 7, no. 1, 2015, pp. 1958–1962., doi:10.2174/1874444301507011958.

## 10 Appendix

### Appendix A. Evaluation Metrics

**Figure 1**

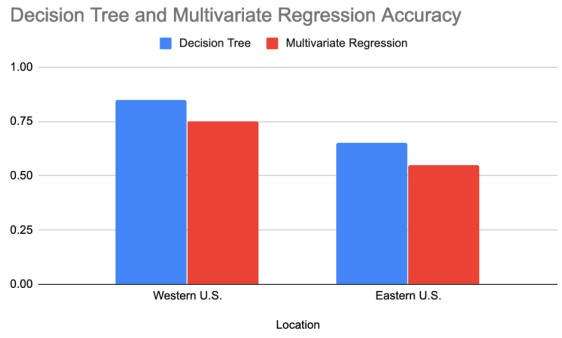


**Figure 2**

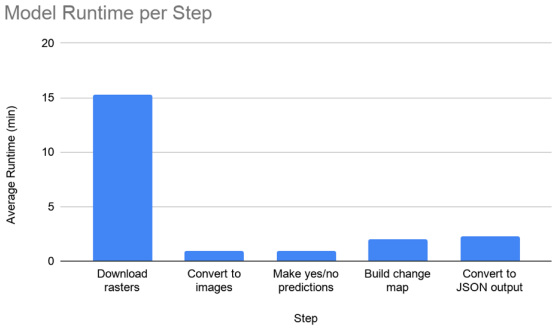


**Figure 3**

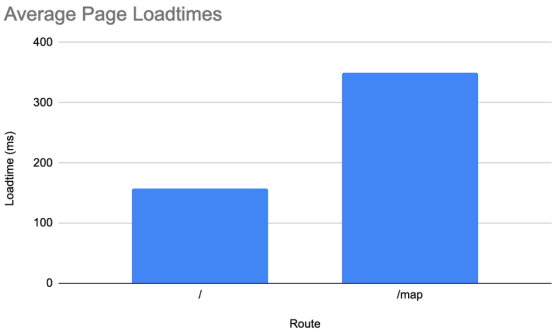


**Figure 4**

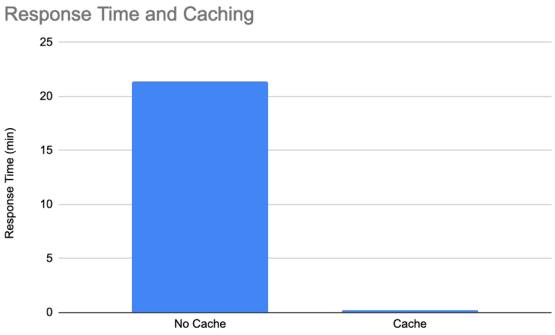


**Figure 5**