Using Random Forest to Classify Natural Landscape Disturbances in OLYM, MORA, and NOCA

Report

Dan Wexler, John Boetsch, and Natasha Antonova

4/19/2024

----------------------------------------------------------------------------------------------------

Table of Contents

1. Introduction
2. Methods
   1. LandTrendr
   2. Assembling Predictor Variables
      1. ArcGIS Pro
      2. Google Earth Engine
   3. Variable Selection
   4. Model Training
      1. Cleaning Data
      2. Train-Test Split
      3. Training
   5. Model Testing
      1. Confusion Matrix
      2. Model Probability
3. Results
4. Discussion
5. Conclusion
6. References

----------------------------------------------------------------------------------------------------

Document Description

This report documents the process of creating a random forest machine learning model to classify natural landscape disturbances in Olympic, Mount Rainier, and North Cascades National Parks, as well as the reasons for doing so and some preliminary results. It is focused mostly on the *reasoning* that went in to creating the model, not *how* to run the model as a user. For step-by-step instructions on how to run the model, see the standard operating procedure (SOP).

1 Introduction

Landscape disturbances are short-term events that modify the land cover of a specific area, usually by reducing the vitality or quantity of that area’s vegetation. Many landscape disturbances are natural, such as avalanches, tree blowdowns, and fire, while others are anthropogenic, such as clearing a stand of trees for the development of a building. Regardless of origin, landscape disturbances can drastically alter the character of a landscape and are important to understand for several reasons.

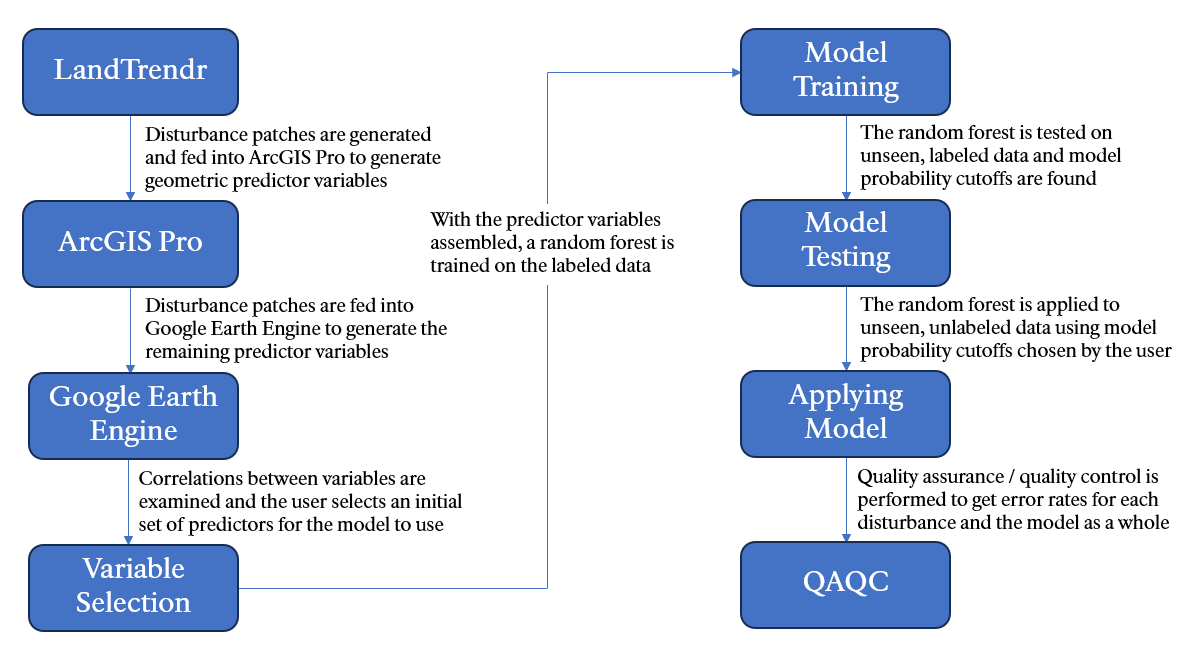
1. Landscape disturbances can change the ecology of a region. Succession is the process through which new plant communities take over land that was previously occupied by another community, and landscape disturbances help facilitate this process. If a forested area that is dominated by tree species A is disturbed by an avalanche, tree species B may take over after the disturbance. Thus, landscape disturbances help maintain biodiversity.
2. The recovery of a landscape after a disturbance is a good indicator of ecological resilience.
3. Natural landscape disturbances sometimes threaten infrastructure and communities.
4. The frequency, distribution, and severity of natural landscape disturbances may be changing due to climate fluctuations. Natural disturbances are driven by precipitation, temperature, and other processes of the Earth’s climate system, and as the climate continues to change it is likely that disturbances will be affected. The first step to understanding how climate change is impacting different aspects of disturbances is knowing where and when disturbances are occurring.

Olympic, Mount Rainier, and North Cascades National Parks all regularly experience landscape disturbances, but the size and remote nature of the parks makes it difficult to track change from the ground. To help solve this issue, the North Coast and Cascades Network (NCCN) of the National Park Service has partnered with Oregon State University to employ a satellite-based disturbance detection algorithm. The method is called LandTrendr (Landsat-based detection of trends in disturbance and recovery) and uses Landsat satellite imagery to detect and flag abrupt changes in vegetative cover on a pixel-by-pixel basis ([1] Kennedy et al., 2010). Once disturbed pixels have been identified, they are grouped into “patches” if they are spatial neighbors and were disturbed in the same year. Each of these patches represents a distinct event.

These patches tell us where disturbances have taken place, but not what type of disturbance occurred. To fully understand what is taking place in the parks, it is important to know whether each patch is an avalanche, blowdown, fire, or some other type of disturbance. One approach to solving this is to look at satellite or other aerial imagery and hand-label each patch, but this can be incredibly time consuming. This paper, building on the work of NCCN’s Landscape Dynamics Monitoring Protocol, demonstrates how a random forest machine learning model ([2] Breiman, 2001) can be used to accurately label a significant portion of the LandTrendr disturbance patches. A novel approach for assessing the random forest’s voting output is introduced.

2 Methods

This section walks through the steps of the modeling process, from generating predictor variables to training and testing the random forest to applying the model on unseen data (Figure 1). A section of the report is devoted to each part of the process.



*Figure 1: A flowchart detailing the process of training, testing, and applying a random forest machine learning model to unseen disturbance patches.*

2.a LandTrendr

The first step in this process is generating the disturbance patches using LandTrendr. There are multiple “knobs” that can be turned on LandTrendr, such as the minimum mapping unit (the smallest number of disturbed pixels that can form a disturbance patch) and the adjacency rule (whether diagonally adjacent pixels are considered neighbors for the purpose of grouping pixels into patches). For this report, the minimum mapping unit was set to five pixels and diagonally adjacent pixels were considered valid neighbors. The algorithm was run on Landsat imagery from 1987 to 2023, and the normalized burn ratio (NBR) spectral index was used to identify change. For this time period, LandTrendr identified 40134, 115372, and 103783 disturbances for Mount Rainier, Olympic, and North Cascades National Parks, respectively.

2.b Assembling Predictor Variables

Characteristic qualities of each patch, known as predictor variables, are used to train the random forest model and help the trained model label unseen patches. 72 unique predictor variables were used in this study, and we assigned each predictor into one of four categories: spectral, landscape, geometric, and space/time. Spectral predictors refer to those that measure the reflectance of the disturbance patch and include different spectral indices at different periods of time. For example, the tasseled cap greenness of each patch is measured one, three, seven, and fifteen years after the disturbance. Landscape predictors get at the underlying nature of the terrain on which the disturbances occur. These predictors include elevation, slope, transformations of aspect, and others. Geometric predictors refer to the shape of the patches and include variables such as perimeter and area. Finally, space/time predictors include those that indicate where and when disturbances take place, such as the year and the latitude. All spectral predictors are generated by the LandTrendr algorithm, and all geometric predictors are calculated in ArcGIS Pro. The remaining predictors are calculated using a mix of ArcGIS Pro, LandTrendr, and Google Earth Engine (GEE). See the “Predictors.xlsx” file in the “documentation” folder of the GitHub repository for more information about each predictor variable used in this study.

2.b.i ArcGIS Pro

ArcGIS Pro is used to calculate the geometric predictor variables for each disturbance patch, as well as the topographic convergence index (TCI) images for each park (and the streams layer?). The geometric predictor variables are calculated using the Zonal Geometry tool in the Spatial Analyst toolbox. The TCI images were calculated using (?).

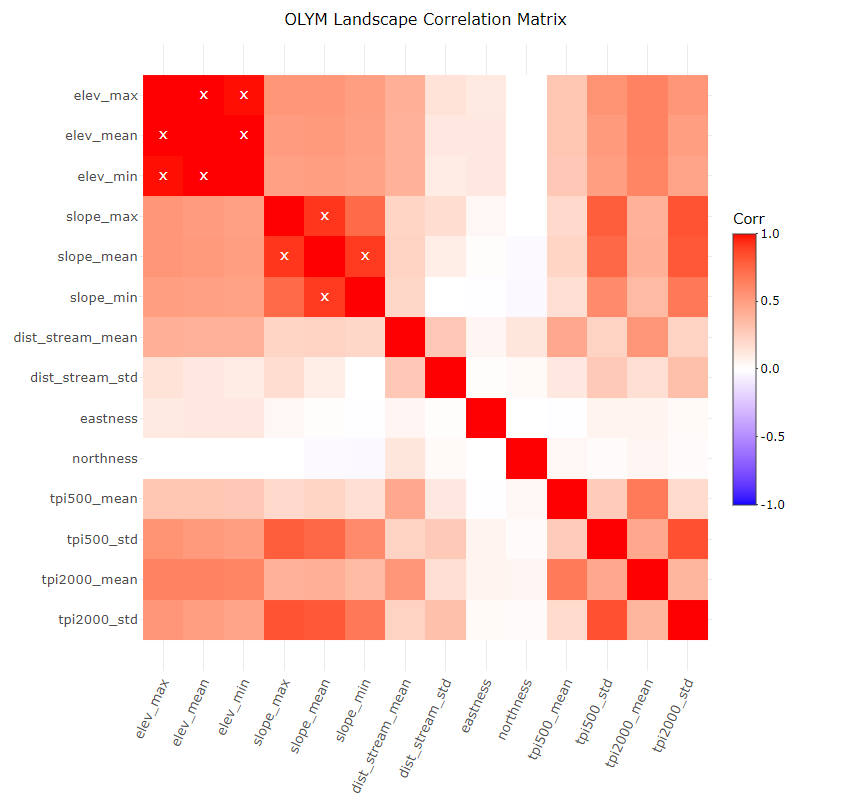
2.b.ii Google Earth Engine

GEE is used to calculate the remaining predictor variables. Many of these remaining predictors could theoretically be calculated using ArcGIS Pro, but the code-based nature of GEE allows for a more flexible and hands-off approach. The geometric predictor variables and TCI images are calculated in ArcGIS Pro instead of GEE because there is not a straightforward way of doing so using GEE. The landscape predictors calculated in GEE are found by sampling an underlying image (e.g., an elevation map) over the extent of each disturbance polygon. Many of these images are derived from a USGS elevation map of the Unites States ([3] USGS).

2.c Variable Selection

Once all the predictor variables have been calculated for each disturbance patch, an initial variable selection process is carried out with the goal of reducing correlated pairs of variables. For example, two predictor variables used in this study are the maximum and minimum elevation of a patch. If these two variables are highly correlated, does the model need to be trained on both? While the accuracy of a random forest model is not strongly affected by the presence of correlated variables, the interpretability of the model is weakened ([4] Genuer et al., 2010). The random forest package in R measures the “importance” of each variable used to train the model, which is based on the decrease in model accuracy that would occur if that variable were removed from the model. If maximum and minimum patch elevation are correlated and are included in the model, it is likely that the importance of the elevation variables would be underscored because the importance is split between these two variables. If one of the two variables is kept and the other is removed, the importance of elevation will become more obvious.

The selection process involves looking at pairs of correlated variables within each predictor category and removing variables one at a time until there are only a few or no correlated pairs remaining. A correlation matrix is used to assist this process (Figure 2). The selected variables are the ones that will be available to the model during training.



*Figure 2: A correlation matrix for Olympic National Park’s landscape variables that is used to assist the variable selection process. The x’s indicate pairs of variables that have a correlation above a user-set cutoff, in this case 0.85.*

2.d Model Training

With the predictors calculated and a parsimonious set of variables assembled, it is time to train the model. This study uses the ‘randomForest’ package in R, an implementation of the random forest algorithm, to build the model ([5] Liaw and Wiener, 2022). The training process is broken down here into three parts: cleaning and filtering the data based on user-set variables, performing the train-test split, and training the model itself.

2.d.i Cleaning Data

There are several variables the user can modify that affect how the random forest is trained, and this section will briefly discuss some of those effects. For more specific information on each of the modifiable variables, see the SOP.

The first major decision that the user must make is what disturbances classes to model on. The table below (Table 1) breaks down the distribution of labels for each park.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mount Rainier | Olympic | North Cascades |
| Annual Variability | 3697 | 361 | 16229 |
| Avalanche | 96 | 415 | 932 |
| Blowdown | 96 | 214 | 43 |
| Clearing | 16 | 73 | 27 |
| Defoliation | 1648 | 32 | 2005 |
| Development | 1 | 5 | 7 |
| Fire | 80 | 131 | 1594 |
| Ice Damage | 0 | 2 | 0 |
| Mass Movement | 38 | 193 | 338 |
| Post Avalanche | 0 | 4 | 10 |
| Post Blowdown | 1 | 0 | 0 |
| Post Clearing | 3 | 0 | 6 |
| Post Defoliation | 0 | 0 | 2 |
| Post Fire | 80 | 74 | 523 |
| Post Mass Movement | 0 | 2 | 2 |
| Riparian Change | 351 | 989 | 313 |
| Water | 3 | 0 | 0 |
| Total | 6110 | 2495 | 22031 |

*Table 1: Frequency distribution of the different disturbance types across the three parks.*

The main consideration when picking what disturbance classes to include was whether there were enough labeled examples of a specific class such that the model would be able to learn and label accurately. For example, there are very few patches labeled as “Development” across all three parks, so this class is left out. The classes selected for this study are annual variability, avalanche, blowdown, clearing, defoliation, fire, mass movement, and riparian change.

The next decision is whether to lump the “post” disturbance classes in with their parent class. The post classes are likely similar across predictor variable space to their parent classes, so it might make sense to combine the classes and bolster the size of the training sets (most of the post classes, Post Fire being the exception, have too few examples to model on separately anyways). For this study, all the post classes were combined with their parent classes. A separate flag is whether to relabel Water patches as Annual Variability, and for this study that was done for each park.

Another flag is whether to drop the disturbance patches from 1987, the lower bound of the LandTrendr run, from the study. The reason for doing this is that, due to the nature of LandTrendr, the disturbance patches labeled as having taken place in 1987 may have taken place a year or two before. Due to this discrepancy, patches from 1987 were left out of this study. The table below (Table 2) shows the breakdown of the disturbances for each park after these filtering operations were performed.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mount Rainier | Olympic | North Cascades |
| Annual Variability | 3029 | 332 | 13439 |
| Avalanche | 96 | 419 | 942 |
| Blowdown | 97 | 214 | 43 |
| Clearing | 16 | 71 | 25 |
| Defoliation | 1648 | 32 | 2005 |
| Fire | 159 | 197 | 2099 |
| Mass Movement | 38 | 195 | 339 |
| Riparian Change | 349 | 980 | 313 |
| Total | 5432 | 2440 | 19205 |

*Table 2: Frequency distributions of the different disturbances after filtering and cleaning.*

2.d.ii Train-Test Split

A train-test split is a common practice in machine learning that divides the labeled data into two sets, a training set and a testing set. The training set is used to train the machine learning model, and the testing set, which the model never sees during training, is used to test the model’s accuracy. In this study, the percent of the data that is used for training is a variable that can be set by the user, and for these runs of the model was set to 0.7, or 70%. This percentage is not applied to all the disturbance patches uniformly, however, as some disturbance classes are much larger than others. If 70% of the disturbances in North Cascades were randomly selected for training, it is likely that no patches from the Blowdown or Clearing categories would make it into the training set. Thus, stratified sampling is performed in which each disturbance class is sampled individually to ensure that all classes are present in the final training set.

There are a few other nuances to this process. Although stratified sampling ensures that each disturbance class is present in the training set, the imbalances in the data would still be reflected. Again, for North Cascades, most patches in the training set would be those from the Annual Variability category. Stark imbalances in the training set can lead to the random forest performing well on classes with the most examples and poorly on those with fewer, which is not necessarily desirable ([6] Sinha et al., 2019). To counteract this, the number of examples from the larger classes in the training set is limited using the equation below (Equation 1). The size of the more prevalent disturbance classes is limited by the size of the disturbance class with the fewest examples and a “balance multiplier” chosen by the user.

Training set size limit = (size of smallest disturbance class \* split) \* balance multiplier

*Equation 1: The formula for calculating the maximum number of patches of a certain disturbance type in the training set. Split refers to the percentage of patches from each class allowed in the training set, which in this case is 0.7. The balance multiplier is an integer.*

In this study, the balance multiplier was arbitrability chosen to be five. A smaller number would better balance the data but severely limit the number of patches used for training. A larger number would imbalance the data further but allow the larger disturbance classes to have more training examples. The latter option may be desirable in certain situations. For example, if Annual Variability is the most common disturbance class in a certain park, it may make sense not to limit the number of Annual Variability patches allowed in the training set. While this imbalance may cause the disturbance classes with fewer examples to be more poorly labeled, a greater number of patches overall may be more accurately labeled, and more time will be saved. The user should adjust the multiplier according to their needs. The final restriction on the training set is that, for each disturbance class, no two patches are allowed to be within 200 meters of each other. This ensures that the training set is spatially diverse, as disturbance patches that are near each other are likely to be similar across the predictor variable space.

Stratified sampling is also performed to build the testing set, but the disturbance classes are only balanced if the user chooses to balance them. An imbalanced testing set will not affect the performance of the model in any way, but it may affect how accuracy of the model is interpreted. One reason to not balance the testing set data is that, for disturbance classes with more examples, the omission error rate will likely be more accurate. However, commission error becomes less meaningful when the testing set is not balanced. For this study, the testing set is not balanced. The testing set is also not subjected to the 200-meter distance rule.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Mount Rainier | | Olympic | | North Cascades | |
|  | Train | Test | Train | Test | Train | Test |
| Annual Variability | 55 | 2974 | 110 | 222 | 90 | 13349 |
| Avalanche | 55 | 41 | 110 | 309 | 90 | 852 |
| Blowdown | 55 | 42 | 110 | 104 | 30 | 13 |
| Clearing | 11 | 5 | 50 | 21 | 18 | 7 |
| Defoliation | 55 | 1593 | 22 | 10 | 90 | 1915 |
| Fire | 55 | 104 | 110 | 87 | 90 | 2009 |
| Mass Movement | 27 | 11 | 110 | 85 | 90 | 249 |
| Riparian Change | 55 | 294 | 110 | 870 | 90 | 223 |

*Table 3: The sizes of the training and testing sets for each park.*

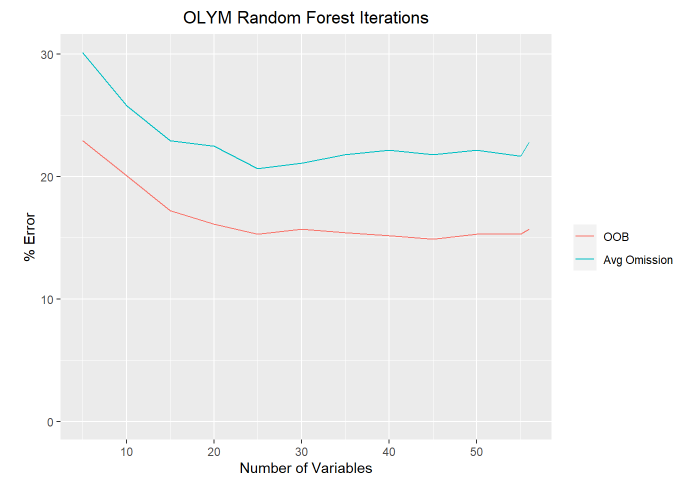
2.d.iii Training

Random forest is a machine learning algorithm that uses an ensemble of decision trees to make classifications or perform regression (this study deals with classification). The decision trees are randomly created, and each node in the tree splits on one or more of the predictor variables that are provided to the model. For a single disturbance patch, each tree will classify the patch as being a certain disturbance type, and the class with the most tree “votes” is the winning classification.

A useful feature of random forest is that one can determine which predictors are the most important, or in other words, which variables best help the model parse the data and make accurate classifications. This importance is defined as the decrease in model accuracy that occurs if a specific variable is left out of the model. The higher the mean decrease in accuracy, the more important the variable. It is important here to briefly explain how these accuracy changes are measured during the training process.

Another useful feature of random forest is that, when training a model, an estimate of the classification error, called out-of-bag (OOB) error, is produced. Each decision tree in the random forest is built from a random selection of the training data, in this case disturbance patches, with replacement. The disturbance patches that were not used to train that specific tree will be fed through that tree and a classification will be made. More broadly, each disturbance patch is fed through every tree that did not include it during the building process and a majority classification is found. The OOB error is the classification error across all the disturbance patches, calculated by feeding the disturbance patches through trees they were not a part of creating. Thus, during the training process, the model can calculate the decrease in OOB error that occurs when a variable is removed from the random forest.

The first step in the training process for this study is creating five random forests and, for each predictor variable, averaging the mean decrease in accuracy across the different models and generating a ranked list of the most important variables. See the Appendix for ranked lists of the variables for each park (Appendix Figures 1-3). The top five variables are then selected, a new random forest trained using only those variables, and the OOB error for that model is found. Then, the five next-best variables are added, and a new model is trained with the top ten variables. This process is repeated until all the variables are included in the model, and the random forest with the lowest OOB error is selected. This is the model that will be tested and used for labeling. The reasoning behind this iterative approach is that, in order to create a more interpretable model, it is better to train on fewer predictor variables. If there are predictors that do not contribute to the model’s ability to classify disturbance patches, there is no reason to include them in the model. The following chart shows how the OOB error changes as the random forest is trained with an increasing number of the most important variables (Figure 3).



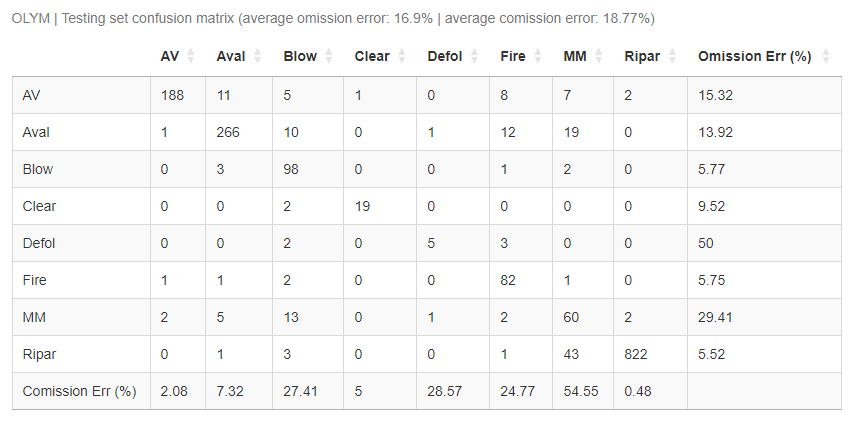
*Figure 3: A graph for Olympic National Park showing how the OOB error changes as a random forest model is trained with progressively more important variables. The average omission error for each iteration is also shown, which is defined as the classification error for each disturbance class averaged over the number of disturbance classes.*

2.e Model Testing

At this point, the final random forest has been selected, and the model’s abilities will be evaluated using the withheld testing data.

2.e.i Confusion Matrix

Each disturbance patch in the testing dataset receives a disturbance label from the model which is then compared against the actual label, and omission and commission error rates are calculated for each disturbance type. Omission error is the percent of patches that are misclassified by the model and is the primary metric that will be used to evaluate the model. Commission error is, for each disturbance class, the percent of labels that were incorrectly attributed to that class. To further clarify this distinction, imagine just the avalanche disturbance class. If there are 100 actual avalanche patches and 90 of those 100 patches are labeled as avalanches, the omission error rate would be 10/100 = 10%. Then, if across the other disturbance types 20 patches were incorrectly labeled as avalanches, the commission error rate would be 30/(90+30) = 25%. The matrix below shows the classification breakdown for Olympic National Park’s testing set with omission and commission error rates for each disturbance type (Figure 4).



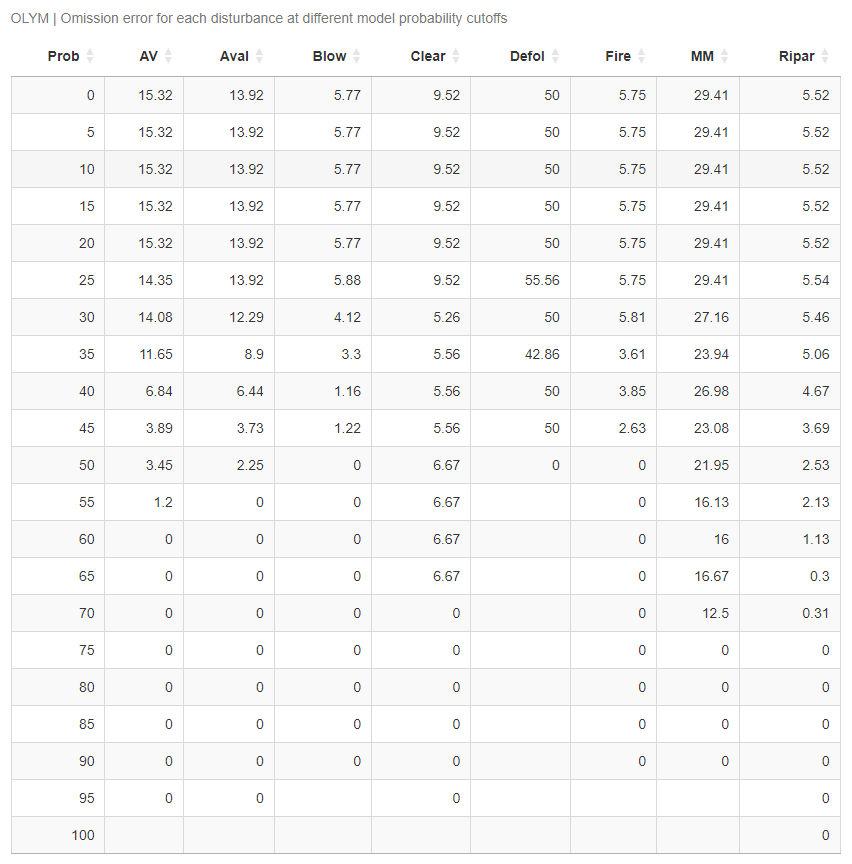
*Figure 4: A confusion matrix for Olympic National Park’s testing dataset. The rows represent the actual disturbance labels, and the columns represent the labels generated by the model. Omission error for each disturbance class is shown on the right, and commission error is shown on the bottom. This study focuses on omission error.*

If the model were now deployed on new, unseen data, the omission error rates calculated using the testing data shown in the confusion matrix are the error rates that would be expected. Is it possible to do better?

2.e.ii Model Probability

A random forest makes a classification by picking the disturbance type that receives the most tree votes. The distribution of these votes will be defined as the model probability. That is, if 40% of the trees in a forest vote that a disturbance patch is an avalanche, that patch has a 40% model probability of being an avalanche. It is important to note that this is not a true statistical probability.

Is it possible that the omission error rate is lower for classifications made at higher model probabilities? If the forest is more “confident” in a certain classification, does it have a lower chance of being wrong? It is found that, across disturbance types, the omission error decreases as the model probability increases. In other words, classifications made more confidently by the random forest are more likely to be correct. Error rates for each disturbance type at different model probability thresholds are shown in the image below (Figure 5).



*Figure 5: A chart showing omission error rates for each disturbance type at different model probability thresholds for Olympic National Park. The model probability column indicates the tree voting threshold imposed to calculate the omission error rate. For example, the sixth row of the table is showing omission error rates for each disturbance using only classifications where greater than 25% of the trees voted for the winning label.*

It follows that, to improve the random forest’s performance, a model probability cutoff could be set for each disturbance type. In effect, this would mean accepting or rejecting classifications based on how many trees in the random forest voted for that label. For example, if no probability cutoff is set, one would expect the forest’s omission error rate for patches labeled as annual variability to be 15.32%, the rate shown in Figure 5. However, if a probability cutoff of 60% is imposed, meaning only disturbance patches given the annual variability label by more than 60% of the trees are examined, the expected error rate drops to 0%, which is a large improvement. This process is effectively improving the performance of the model by only accepting confident classifications. The downside of this approach is that fewer disturbance patches will have accepted labels at higher probability cutoffs. Thus, the goal of the user is to choose model probability cutoffs that balances omission error and the percent of the patches that are classified above the cutoff. For each disturbance type within each park, a model probability cutoff is chosen (Table 4).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mount Rainier | | | Olympic | | | North Cascades | | |
|  | Cutoff | Error | % Acp | Cutoff | Error | % Acp | Cutoff | Error | % Acp |
| AV | 0 | 3.67 | 100 | 45 | 3.89 | 81.08 | 0 | 4.66 | 100 |
| Aval | 45 | 3.33 | 73.17 | 45 | 3.73 | 77.99 | 65 | 4.9 | 50.35 |
| Blow | 30 | 2.63 | 90.48 | 0 | 5.77 | 100 |  |  |  |
| Clear |  |  |  | 0 | 9.52 | 100 |  |  |  |
| Defol | 0 | 4.39 | 100 |  |  |  | 50 | 4.72 | 81.88 |
| Fire | 0 | 6.73 | 100 | 0 | 5.75 | 100 | 65 | 3.81 | 53.56 |
| MM |  |  |  |  |  |  |  |  |  |
| Ripar | 35 | 3.27 | 93.54 | 0 | 5.52 | 100 | 65 | 4.97 | 72.2 |

*Table 4: The chosen model probability cutoffs for each disturbance type within each park, as well as the associated omission error rates and percent of accepted labels. The error rates and percent of accepted labels help inform what probability cutoff is chosen and hopefully reflect the true behavior of the random forest model on unseen data.*

3 Results

A separate random forest model was trained for each park, and within each park model probability cutoffs were found for each of the eight disturbance types. These models were then applied to new, unseen disturbance patches and the labeling results were subjected to the probability cutoffs shown in Table 4. For Mount Rainier, no clearing or mass movement labels were accepted. For Olympic, no defoliation or mass movement labels were accepted. For North Cascades, no blowdown, clearing, or mass movement labels were accepted. The following charts show the labeling result for each park.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Disturbance | # Total Labels | # Accepted | % Accepted | Expected Error % | # Secondary Labels |
| AV | 1364 | 1364 | 100 | 3.67 | 98 |
| Aval | 467 | 125 | 26.77 | 3.33 | 0 |
| Blow | 303 | 240 | 79.21 | 2.63 | 10 |
| Clear | 4 | 0 | 0 |  | 0 |
| Defol | 3959 | 3959 | 100 | 4.39 | 57 |
| Fire | 1200 | 1200 | 100 | 6.73 | 90 |
| MM | 292 | 0 | 0 |  | 0 |
| Ripar | 503 | 443 | 88.07 | 3.27 | 0 |

*Table 6: The labeling results from Mount Rainier National Park.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Disturbance | # Total Labels | # Accepted | % Accepted | Expected Error % | # Secondary Labels |
| AV | 12344 | 10225 | 82.83 | 3.89 | 0 |
| Aval | 1733 | 831 | 47.95 | 3.73 | 0 |
| Blow | 1495 | 1495 | 100 | 5.77 | 1091 |
| Clear | 190 | 190 | 100 | 9.52 | 15 |
| Defol | 238 | 0 | 0 |  | 0 |
| Fire | 1140 | 1140 | 100 | 5.75 | 911 |
| MM | 616 | 0 | 0 |  | 0 |
| Ripar | 1765 | 1765 | 100 | 5.52 | 209 |

*Table 7: The labeling results from Olympic National Park.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Disturbance | # Total Labels | # Accepted | % Accepted | Expected Error % | # Secondary Labels |
| AV | 14104 | 14104 | 100 | 4.66 | 3079 |
| Aval | 3876 | 777 | 20.05 | 4.9 | 0 |
| Blow | 3285 | 0 | 0 |  | 0 |
| Clear | 119 | 0 | 0 |  | 0 |
| Defol | 3869 | 2607 | 67.38 | 4.72 | 0 |
| Fire | 6962 | 1338 | 19.22 | 3.81 | 0 |
| MM | 1938 | 0 | 0 |  | 0 |
| Ripar | 982 | 296 | 30.14 | 4.97 | 0 |

*Table 8: The labeling results from North Cascades National Park.*

The following table (Table 9) shows what percent of each park’s disturbance patches were classified with an accepted label, and how many remaining disturbances must be labeled by hand.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mount Rainier | Olympic | North Cascades |
| # Accepted Labels | 7331 | 15646 | 19122 |
| Total # Labels | 8092 | 19521 | 35135 |
| % Accepted | 90.6% | 80.15% | 54.42% |
| # Unaccepted | 761 | 3875 | 16013 |

*Table 9: The breakdown of accepted labels for all three parks.*

In total, 42099 out of 62748 patches received an accepted label, or 67.1%. This leaves 20649 patches to label by hand. Of these 20649 remaining patches, 3.7% come from Mount Rainier, 18.8% come from Olympic, and 77.5% come from North Cascades. The model developed for Mount Rainier performed the best, followed by Olympic, with North Cascades leaving the most unaccepted labels both by magnitude and percent.

4 Discussion

Based on the results found by applying the model to the testing data, thresholding the model’s labels with probability cutoffs is an effective way to artificially reduce the random forest’s error rate and increase the model’s performance. The tradeoff is, of course, fewer disturbance patches receiving accepted labels. Large gains in the overall number of accepted patches could be made by improving the model for North Cascades National Park, as 77.5% of all the unaccepted labels are from that park. Some ideas for improving this model in the future are increasing the number of trees in the random forest, setting the balance multiplier to a higher integer, and introducing new predictor variables that better describe the disturbance types for which the model did not perform well.

The immediate next step for this project should be performing a QAQC on a subset of the disturbance patches labeled by the model, for each disturbance type within each park. This QAQC would involve looking at the disturbance patches in ArcGIS Pro and recording whether patch labels match the actual disturbance type. This QAQC would allow a greater confidence in the effectiveness of the model and allow for accurate error reporting. This study suggests that a sample of 50 patches for each disturbance type for each park may be sufficient for estimating error rates.

With more time, there are multiple avenues of this project that could be pursued, one being determining a minimum distance between disturbances of the same type in the training set. For this study, this distance was set to 200 meters, but this was chosen arbitrarily based on semivariograms for landscape change spectral variables. Semivariograms should be produced for each predictor variable in each park in order to determine how underlying predictors change over space. The goal of this process is to reduce spatial autocorrelation and ensure that the training data is spatially representative of all the disturbances.

The goal of the QAQC is to determine if the error rates found by testing the model on the testing data are close to the actual error rates of applying the model to new, unseen data. It would also be interesting to see if the expected percent of patches that receive an accepted label changes between the testing set and new data. Hopefully the testing set is representative of new data, but it would be good to check.

Another avenue to pursue is finding ways of reducing the variability between different models. For a single park, different models can produce different omission error rates, model probability cutoffs, and labeling results. While some variability is to be expected, it would be good to quantify this change and try to reduce it as much as possible. One of the most noticeable ways in which models vary is how they perform on disturbance classes with a middling number of disturbance patches. The largest disturbance classes consistently model well, and the smaller disturbance classes consistently model poorly, but the classes in the middle often perform well or do not depending on the model. One way to analyze this variability would be to artificially increase the size of these middle disturbance classes by including multiple instances of a single disturbance patch in the training set in order to bolster its size. It could be determined whether there was a set size at which the variability is minimal. Another, perhaps better way of accomplishing this analysis would be to artificially decrease the size of a large disturbance type and see where variability starts to increase. A second way of reducing the variability between models would be to pursue an ensemble approach ([7] Jaeger et al., 2023) in which multiple random forests are trained and the results are aggregated to produce a label for each disturbance patch. Because the variability is on a model-by-model basis, an ensemble approach might go a long way towards reducing this variability.

Built into the training code is an option that, if turned on, trains a new random forest model without performing a train test split after model probability cutoffs have been determined using the first model and the testing data. The idea is that the smaller disturbance classes may benefit from not having some of their data withheld in the split, and that the model probability cutoffs determined by the first model would still be applicable to the new model. While it is generally not smart to train a new model and make assumptions about it based on a separate model, the validity of this approach could be determined when a QAQC is performed on new labels.

It would also be interesting to evaluate the performance of a single model for all three parks. While each park is unique and likely occupies a different area of the variable space, combining the data from the three parks together would bolster the number of training examples for the smaller disturbance classes and perhaps lead to better performance for those classes.

Finally, an analysis on the frequency, severity, and distribution of the different disturbances should be performed. This report details the modeling process, but the long-term goal is to better understand landscape disturbances and how they are being affected by climate change and other factors. At this point, it might be good to report model error based on the physical area that the disturbance cover, rather than the number of patches that are classified correctly.

5 Conclusion

This report details the process of creating random forest machine learning models to label landscape disturbances in Mount Rainier, Olympic, and North Cascades National Parks. The models for Mount Rainier and Olympic are effective at labeling most of the disturbance patches at a low error rate, and the North Cascades model can label some patches accurately but not nearly as many as in the other two parks. A novel way of analyzing the random forest voting output to artificially lower the model’s omission error rate is introduced. With more time, a comprehensive QAQC would be performed to confirm the effectiveness of the model, and other larger changes would be implemented to reduce the variability of the results from run to run and improve model performance for the more infrequent disturbance types.

6 References

[1] Kennedy, Robert E., et al. “Detecting trends in Forest Disturbance and recovery using yearly Landsat Time Series: 1. Landtrendr — temporal segmentation algorithms.” *Remote Sensing of Environment*, vol. 114, no. 12, 15 Dec. 2010, pp. 2897–2910, https://doi.org/10.1016/j.rse.2010.07.008.

[2] Breiman, Leo. “Random Forests.” *Machine Learning*, vol. 45, no. 1, Oct. 2001, pp. 5–32, https://doi.org/10.1023/a:1010933404324.

[3] U.S. Geological Survey, 3D Elevation Program 10-Meter Resolution Digital Elevation Model.

[4] Genuer, Robin, et al. “Variable selection using random forests.” *Pattern Recognition Letters*, vol. 31, no. 14, 15 Oct. 2010, pp. 2225–2236, https://doi.org/10.1016/j.patrec.2010.03.014.

[5] Liaw A, Wiener M (2002). “Classification and Regression by randomForest.” R News, **2**(3), 18-22. [https://CRAN.R-project.org/doc/Rnews/](https://cran.r-project.org/doc/Rnews/).

[6] Sinha, Parmanand, et al. “Assessing the spatial sensitivity of a random forest model: Application in gridded population modeling.” *Computers, Environment and Urban Systems*, vol. 75, May 2019, pp. 132–145, https://doi.org/10.1016/j.compenvurbsys.2019.01.006.

[7] Jaeger, Kristin L., et al. “Predicting probabilities of late summer surface flow presence in a glaciated mountainous headwater region.” *Hydrological Processes*, vol. 37, no. 2, Feb. 2023, https://doi.org/10.1002/hyp.14813.