FireEye

Spas Angelov

June 26, 2024

1. Introduction

In recent years, the exacerbated effects of climate change have become more apparent and damaging to our daily lives. Climate instabilities have fueled a rise in wildfire occurrences which can cause losses to local ecologies, properties, and even human lives. FireEye aims to predict the day-to-day evolution of fires based off of easily accessible satellite data, giving incident responders another tool in their handling of wildfires.

2. Data

Data for this project was collected and analyzed using Google Earth Engine. The project draws inspiration from the paper "Next Day Wildfire Spread: A Machine Learning Data Set to Predict Wildfire Spreading from Remote-Sensing Data" [**1**]. However, the data collection and cleaning processes were independently developed for this project.

Google Earth Engine allows users to access *ImageCollections* which are representations of a group of *Images*. These *Images* are in turn collections geospatial data. An example of an *Image* can be seen in Figure 2.1.

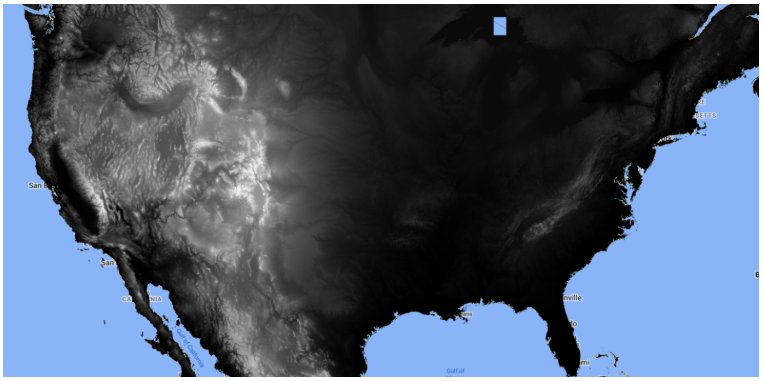

Figure 2.1: Elevation *Image* of United States

Figure 2.1 is an *Image* where each pixel has elevation data in meters. The resolution of this data is around 90m. The visualization is created by setting the lowest elevation pixels black, the highest at white, and then linearly interpolating to create the gradient. Under the hood however, the *Images* can be though of like a 2D array.

For this project, the following *Images* are considered:

- Fire Bit Mask at 1km resolution [**2**]
- Elevation at 90m resolution[**3**]
- Weather at 4km resolution[**4**]
- Drought at 4.5km resolution [**5**]
- Vegetation at .5km resolution [**6**]
- Population Density at 1km resolution [**7**]

Images can contain multiple different bands of information, for example the weather *image* contains bands that have information on precipitation, humidity, temperature, etc... Table 2.1 shows which image each of the selected features are from, and their respective units. Next day fire mask is calculated by taking the fire mask image one day in the future. It's important to note that the fire mask and next day fire mask *image* is a 2-bit mask. The presence of the first bit indicates whether the observation is uncertain due to external weather factors such as clouds. Presence of the second bit indicates that the 1km x 1km region has an active fire.

Feature	**Unit**	**Image Src.**
fire_mask	2 bit mask	Fire[**2**]
fire_mask_next_day	2 bit mask	Fire[**2**]
elevation	meters	Elevation[**3**]
wind_direction	degrees	Weather[**4**]
wind_speed	meters/second	Weather[**4**]
energy_release_component	NFDRS index	Weather[**4**]
burn_index	NFDRS Index	Weather[**4**]
precipitation	millimeters/day	Weather[**4**]
temperature_min	kelvin	Weather[**4**]
temperature_max	kelvin	Weather[**4**]
drought_index	Palmer drought severity index	Drought[**5**]
vegetation	NDVI	Vegetation[**6**]
population_density	persons / km^2	Population Density[**7**]

Table 2.1: Selected Features and Their Units

3. Data Cleaning & Processing

For elevation data, the SRTM dataset [**3**] is just one *image* (ie static through time). This is not the case for other datasets used - many have time components as well. For example, weather and fire data comes in at intervals of 1 day, vegetation comes in at a 16 day interval, and drought is captured every 8 days.

Filtering these *Image Collections* down to individual images is done by collecting the most recent data for a given time stamp. The original study looked at the years 2012-2020, so in an attempt to get a more recent analysis, the chosen dates are between 2020 and 2024. Winter months are excluded from the data export in an effort to save computation time during months where wildfires are less likely. The exported data is also clipped to cover only the area of Colorado.

Images come in different resolution sizes. Like the original study, this project standardizes resolutions to 1km x 1km squares. For data that is higher resolution, this is as simple as taking averages across the chosen area, for lower resolution data, bi-cubic interpolation is used. Images are projected to "CRS: EPSG:2232", to minimize data distortions for Colorado, and then exported as a series of 64km x 64km pixel images encompassing Colorado, where a single pixel represents the 1km x 1km area. The exported data is in the TensorFlowDataset format.

Figure 3.1 is an example of what a 64km x 64km "patch" looks like across the selected features from each *Image*.

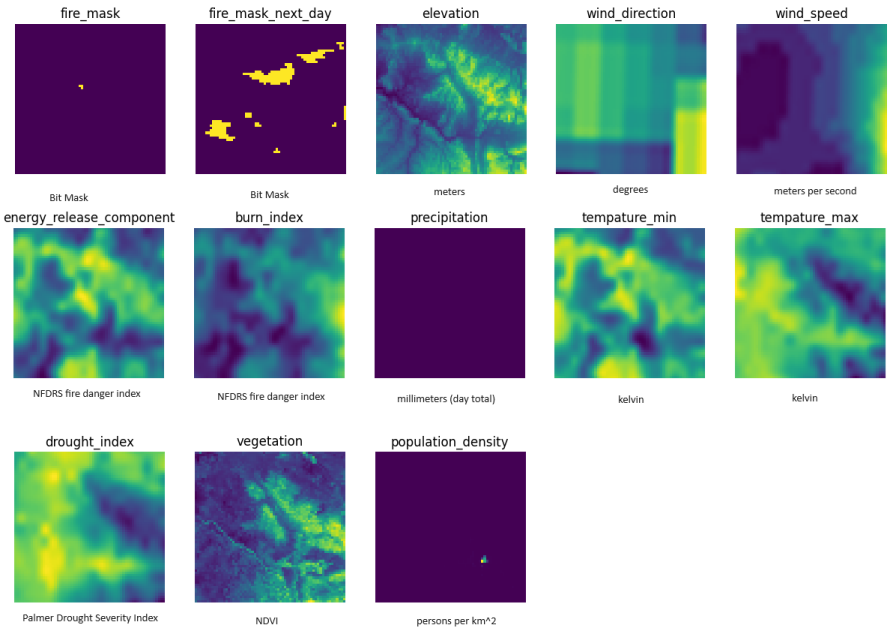

Figure 3.1: 64km x 64km patch over selected features

As is, this data isn't immediately useful. Fires typically start due to external events that aren't captured in this dataset, such as human activity. Prediction of when fires start is

outside of the scope of this project, instead the evolution and spread is what is desired. In order to track fire evolution, the fire mask feature should contain at least one pixel in the 64 x 64 pixel image where there is an active fire. This simple filter reduces the data size from 4GB to around 40MB.

After filtering, this data set is now well made to be ingested into a convolutional neural network to take advantage of the spatial data. However, the class of models this project is exploring are single classifier univariate models, and as such predict only one output variable at a time. In this case, that would mean a 1km x 1km bit pixel of either fire or no-fire. Furthermore, the complete 64 x 64 pixel images across 12 features would result in effectively 49,512 inputs. For a singular prediction, this is far to many features. To simplify the process a bit, for each 64 x 64 pixel image, 3 x 3 pixel subimages are extracted, with only the center pixel for the next day fire mask being predicted.

This sort of processing reduces feature count down to 108. It also gives an opportunity to filter out uncertain fire mask pixels in both the input and the output. Figure 3.2 visualizes how these subimages are created from the larger 64 x 64 patch.

A balanced dataset with roughly equal amounts of fire and no-fire instances on the next day is created. This is done by tuning a random threshold parameter until there are roughly equal fire and no fire examples. After a bit of tuning, the threshold parameter that keeps around 1 out of 450 no-fire examples produces a balanced dataset. The final working dataset is just kept as in a csv file, with 108 feature columns and 1 output column.

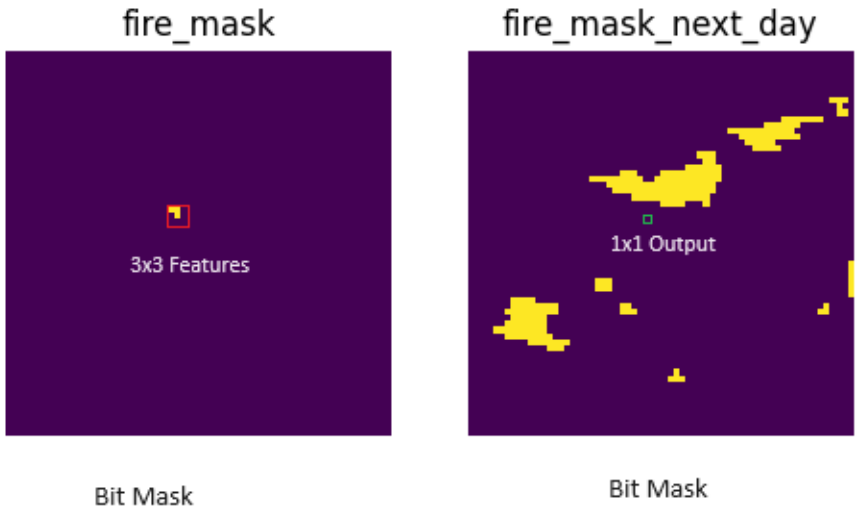

Figure 3.2: Subimage extraction

4. Exploratory Data Analysis

One downside of the data cleaning and processing method used, is that it overly weighs next day fire examples. In addition to this, because the non-fire examples are randomly selected

3x3 pixels over a 64x64 base pixel image, the amount of more ambiguous scenarios is likely reduced. A scenario this could cause is using the center fire mask pixel as the only predictor of the next day fire mask. While this makes sense in practice (something on fire today, will probably be affected tomorrow), the aim is to predict the evolution of this fire over time.

Additionally, since all 12 features are spatial in nature, there is strong correlation between each of the 9 sub features that represent the 3x3 image data for the given base feature.

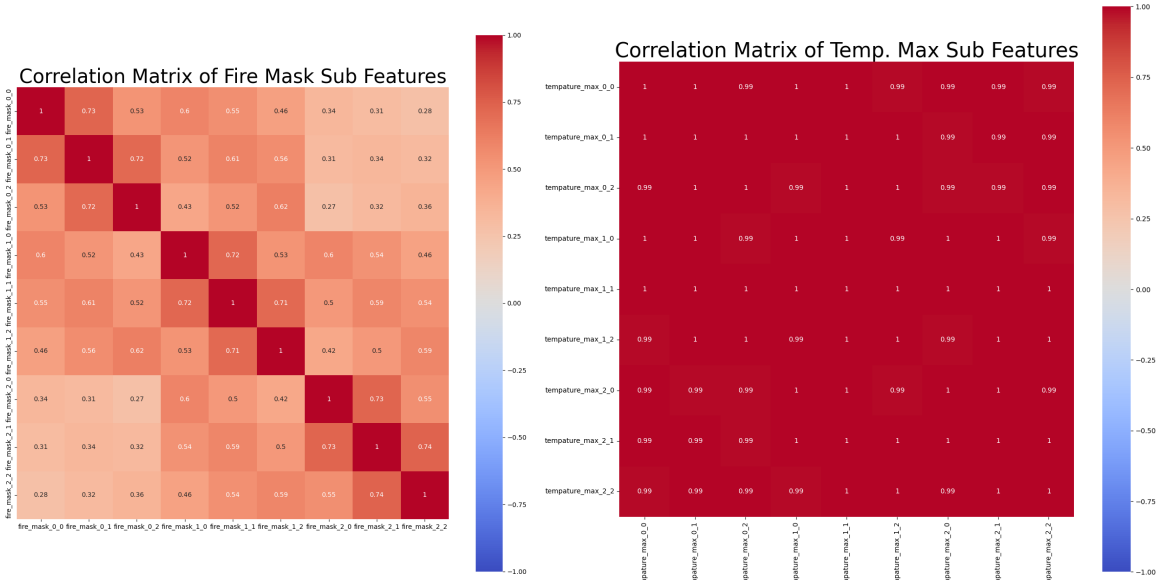

Figure 4.1: Fire Mask Subfeature Correlation(Left) and Temperature Max Correlation(Right) showcasing strong corrolation between subfeatures.

This issue is especially pronounced for data that originally had lower resolution, such as weather data, which was upscaled using bi-cubic interpolation. In this case, the values within a 3km x 3km area are nearly identical.

This suggests that of the 108 features, many of them are redundant. To tackle this each of the subfeatures can be averaged together to reduce the redundancy of the dataset, while also retaining some of the spatial information. The fire mask is the only feature that will stay split up into it's subfeatures. Even though the correlation is fairly high, the spatial data seems important to keep within the model for this feature. Figure 4.2 shows the correlation after averaging.

Besides the intentionally left fire mask subfeatures, this correlation matrix suggests that most features are uncorrelated, and thus not cross redundant. An argument could be made that there is some redundancy for temperature max and min, and the burn index and the energy release component features, however these are left for the model to decide which are most important.

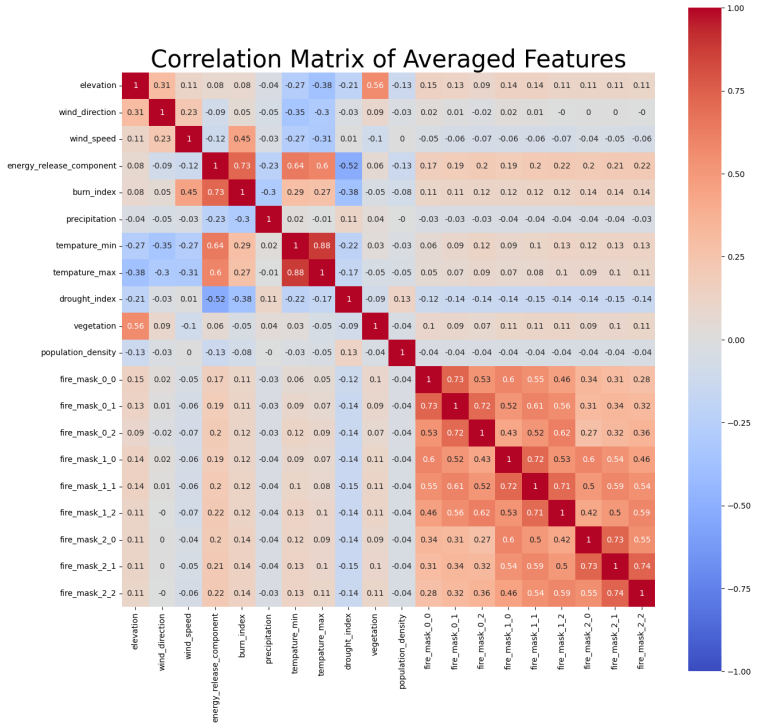

Figure 4.2: Averaged Features Correlation Matrix

5. Models

5.1. Logistic Model & Analysis

Due to the binary classification nature of this problem (with outcomes of either fire or no-fire), a natural selection for model choice is the Logistic Model. Prior to fitting, feature data is scaled to have a mean of 0, and a standard deviation of 1. Fitting is carried out using L2 regularization. From the main dataset, 80% is used to as the training dataset with the remainder as the test set. Running the fit gives fairly high results with a training accuracy of .854 and a test accuracy of .843. Figure 5.1 displays the confusion matrix for the classification.

The simple model has a easier time classifying true positives than true negatives. Despite the decent performance, with 20 features, there is a chance that some are redundant and/or statistically insignificant. Figure 5.2 shows the model summary.

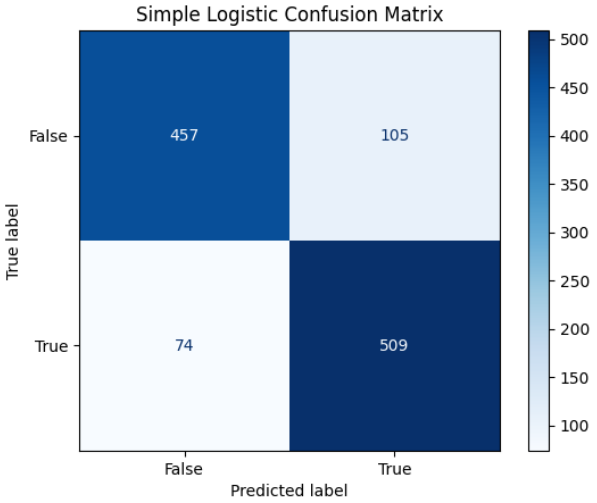

Figure 5.1: Simple Logistic Model - No Feature Selection

```
The summary of the model is as follows;
Optimization terminated successfully.
         Current function value: 0.311859
         Iterations 13
['const', 'elevation', 'wind_direction', 'wind_speed', 'energy_release_component', 'burn_index',
 'precipitation', 'tempature_min', 'tempature_max', 'drought_index', 'vegetation', 'population_density',
  'fire_mask_0_0', 'fire_mask_0_1', 'fire_mask_0_2', 'fire_mask_1_0', 'fire_mask_1_1', 'fire_mask_1_2',
   'fire_mask_2_0', 'fire_mask_2_1', 'fire_mask_2_2']
                           Logit Regression Results
==============================================================================
Dep. Variable:     fire_mask_next_day   No. Observations:                 4576
Model:                          Logit   Df Residuals:                     4555
Method:                           MLE   Df Model:                           20
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                  0.5495
Time:                        20:24:38   Log-Likelihood:                -1427.1
converged:                       True   LL-Null:                       -3168.1
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================================
                                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -21.7676      5.115     -4.256      0.000     -31.792     -11.743
elevation                      0.0022      0.000     12.262      0.000       0.002       0.003
wind_direction                 0.0013      0.001      1.986      0.047    1.68e-05       0.003
wind_speed                     0.3642      0.114      3.189      0.001       0.140       0.588
energy_release_component       0.1205      0.014      8.742      0.000       0.093       0.147
burn_index                    -0.0163      0.013     -1.275      0.202      -0.041       0.009
precipitation                 -1.3467      0.488     -2.762      0.006      -2.302      -0.391
tempature_min                 -0.1784      0.020     -9.060      0.000      -0.217      -0.140
tempature_max                  0.1856      0.020      9.093      0.000       0.146       0.226
drought_index                 -0.2140      0.037     -5.823      0.000      -0.286      -0.142
vegetation                -5.456e-05   3.89e-05     -1.402      0.161      -0.000    2.17e-05
population_density            -0.2108      0.047     -4.531      0.000      -0.302      -0.120
fire_mask_0_0                  2.7715      0.640      4.329      0.000       1.517       4.026
fire_mask_0_1                  1.2082      0.593      2.038      0.042       0.046       2.370
fire_mask_0_2                  2.2288      0.540      4.127      0.000       1.170       3.287
fire_mask_1_0                  1.5893      0.503      3.161      0.002       0.604       2.575
fire_mask_1_1                  0.9301      0.523      1.780      0.075      -0.094       1.954
fire_mask_1_2                  1.5443      0.446      3.461      0.001       0.670       2.419
fire_mask_2_0                  3.1758      0.619      5.128      0.000       1.962       4.390
fire_mask_2_1                  2.1353      0.745      2.868      0.004       0.676       3.595
fire_mask_2_2                  2.2452      0.503      4.462      0.000       1.259       3.232
================================================================================================
```

Figure 5.2: Simple Model Summary

Given the statistical significance of each feature, backward stepwise elimination can be performed to better generalize the model. The processes removes burn_index, vegetation, and notably fire_mask_1_1. Removing burn_index is not too surprising, given that the energy_release_component was correlated with this variable, however vegetation and especially fire_mask_1_1 are two that are surprising. The fire_mask_1_1 feature is the center of the 3x3 current fire mask, and without any analysis seems like it would be a strong predictor of the next day. Perhaps, the fire burns some of its fuel the previous day, making it challenging to tell whether the fire will still be there the next day. Refitting the model after dropping these features gets the following accuracy:

Training Accuracy: 0.8548951048951049
Test Accuracy: 0.8419213973799127

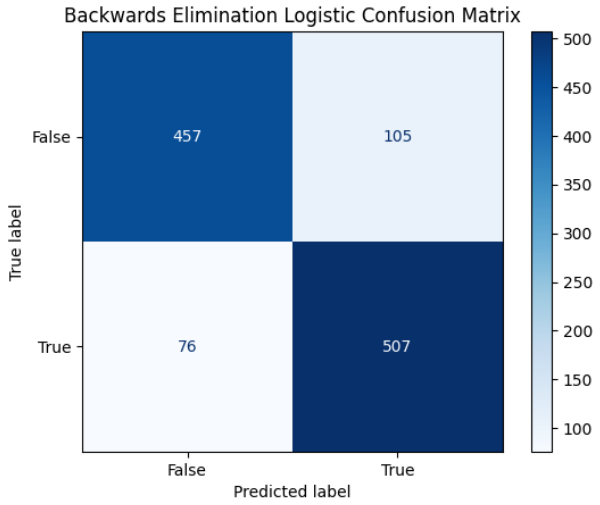

Figure 5.3: Reduced Feature Logistic Model Confusion Matrix

Dropping these 3 features lead only added 2 instances of misclassification for the validation set, however reducing the model complexity means that the variance of the model is reduced as well.

In an attempt to enhance the spatial information provided to the model, new features were created by combining the existing fire mask, wind speed, and wind direction. Wind speed and direction were already part of the model, measured in meters per second and degrees, respectively. The goal was to improve the weighting of the current fire masks based on these factors, providing a pathway towards an interaction that might exist.

The new features were defined as follows:

$firemask_{dy} = firemask \cdot windspeed \cdot \sin(winddirection)$

$firemask_{dx} = firemask \cdot windspeed \cdot \cos(winddirection)$

These new features attempt to encode information about how quickly a fire could move both horizontally and vertically.

However, the resulting model demonstrated a decrease in performance on the testing dataset, despite showing an improvement in accuracy for the training set. This discrepancy indicated overfitting to the training data. Further analysis revealed that most of the p-values for these new features were statistically insignificant. It appears that the wind speed's strength is more critical than the directional information for predicting fire spread. Consequently, these variables were ultimately excluded from the final model.

5.2. Random Forest Model & Analysis

Another model to consider is the Random Forest. Unlike logistic regression, Random Forest models are not sensitive to unscaled data. This is because Random Forests use value thresholds as criteria to make splits, eliminating the need for data scaling.

Fitting the same train dataset as the initial Logistic regression model with a Random Forrest classifier achieves the following training and test accuracies:

Training Accuracy: 1.0
Test Accuracy: 0.965

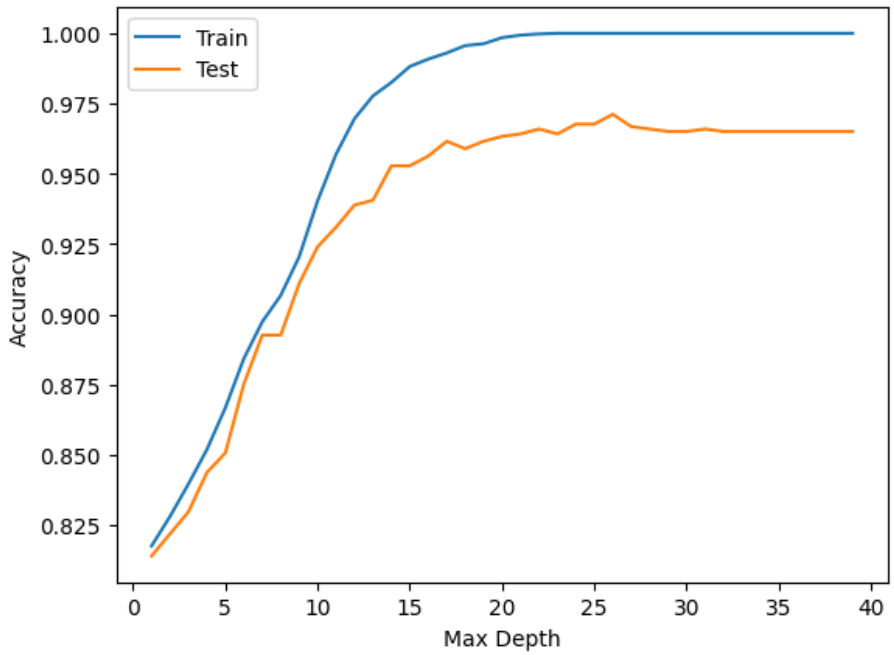

Figure 5.4: Accuracy over a range of Max Depths

A significant drawback of Random Forests is their tendency to overfit when certain hyperparameters are not set. The above results were achieved without explicitly setting any hyperparameters, leading to unlimited maximum depth and a perfect categorization of the training data. To address this issue, the maximum depth parameter can be varied, and the

corresponding training and test accuracy scores can be recorded to find an optimal balance between bias and variance. Generally the aim is to keep the model complexity as low as possible without compromising the train and validation accuracy scores.

As seen in Figure 5.4, the elbow of the graph happens at around max depth set to 19, and for that specific model these are the following scores:

Training Accuracy: 0.996284965034965
Test Accuracy: 0.9615720524017467

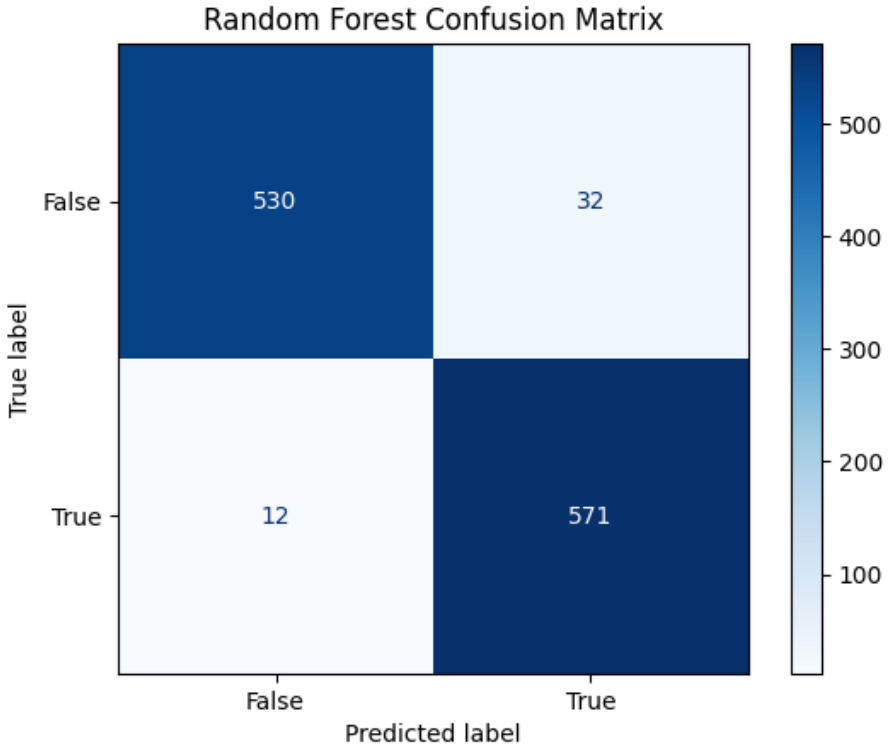

Figure 5.5: Random Forrest Confusion Matrix

Table 5.1 shows the model features ordered in order of how much each individual feature split reduces the Gini impurity.

At first glance, the Logistic and Random Forest models rate feature importance similarly. For example, energy release component is one of the more statistically significant features listed in the Logistic regression and is the most important feature in the Random Forest. There are some differences of course. Burn index does not make it into the final Logistic model at all, despite being fairly highly ranked in the Random Forest model.

Feature	Importance
energy_release_component	0.170833
drought_index	0.115424
elevation	0.111026
burn_index	0.108877
wind_speed	0.055313
wind_direction	0.046511
fire_mask_1_1	0.046265
tempature_min	0.044932
tempature_max	0.042080
fire_mask_2_0	0.039943
population_density	0.039916
vegetation	0.029940
fire_mask_2_1	0.029611
fire_mask_1_2	0.022466
fire_mask_1_0	0.022375
fire_mask_2_2	0.019028
fire_mask_0_0	0.018429
fire_mask_0_1	0.017270
fire_mask_0_2	0.014111
precipitation	0.005650

Table 5.1: Random Forest Feature Importance

6. Results & Conclusion

The final results of both models are listed below:

	Logistic Model	**Random Forest**
Training Accuracy	0.8549	0.9963
Test Accuracy	0.8419	0.9616

Table 6.1: Summary of Model Accuracies

Both models score very high on an accuracy test, which is an appropriate metric given the balanced nature of the input data set. One reason that the Random Forest performed slightly better is because feature interactions are built into the model. This may be especially important for spatial features that would end up augmenting other features. For example, wind speed and direction can change where fire is likely to spread. This interaction is more easily caught with the random forest model, whereas for the Logistic model, entirely new features need to be introduced to account for this. The Logistic model also struggles with large amounts of features, so introducing these spatial interaction terms only really compounds the problem.

The high model accuracy can also be a potential cause for concern, however. It may indicate a deeper problem with how the data was collected. In order to get a balanced dataset (and also a manageabley-sized dataset), uncertain pixels were discarded, all next day fire occurrences were included, and only a random 1 out of 450 non fire samples were included. The way this data is selected introduces inherent bias.

Firstly, in the case of a non-fire pixel, the outcome is very obvious - If no fire mask is present in the previous day's local region, then we expect no fire pixel will be present the next day. This scenario happens in 99% of the predictions where a non fire outcome was determined. In a sense, the no fire outcome is not actually representative of the information trying to be predicted, i.e. if a fire is *is* present, what is the likelihood that it won't spread. Ideally, there would be more ambiguity in this negative case.

This is part of the reason that the fire masks are not rated very high in significance, for example. Each individual pixel out of the 9 does not contain enough information by itself to make a judgment for the next day. However if they are all averaged together, and passed into a Random Forest classifier, the following feature importance list is achieved:

```
                  Feature   Importance
             fire_mask_avg    0.228939
  energy_release_component    0.138060
             drought_index    0.117416
                burn_index    0.117371
...
```

Second, the exclusion of any uncertain pixel information drastically reduces the available dataset. Ideally this scenario is handled through a custom loss function, which doesn't incur

any loss if the output pixel is uncertain. Scikit doesn't support custom loss functions by default, but this could be accomplished Tensorflow, for example.

Finally, given the spatial nature of the original 64x64 pixel patches, a model that takes advantage of the full input and produces a full output is going to be a more efficient way to predict the evolution of wildfires over time. Examples of such models include Logistic Neural Networks and Convolutional Neural Networks.

In conclusion, while both the Random Forest and Logistic Regression models achieve high accuracy, this metric alone does not capture the complexities and potential biases in the data collection process. The high accuracy of both models also hints at a deeper issue: the dataset's inherent bias due to the exclusion of uncertain pixels and the disproportionate sampling of non-fire instances. This approach leads to an over-simplified non-fire outcome, undermining the predictive value of certain features. To address these challenges, future work should explore models that leverage the full spatial context of the data–such as Convolutional Neural Networks–as well as consider frameworks that handle uncertainty more robustly–like custom loss functions in TensorFlow.

References

[1] F. Huot, R. L. Hu, N. Goyal, T. Sankar, M. Ihme, and Y.-F. Chen, "Next Day Wildfire Spread: A Machine Learning Data Set to Predict Wildfire Spreading from Remote-Sensing Data", arXiv preprint, 2021.

[2] Giglio, L., Justice, C. (2021). MODIS/Terra Thermal Anomalies/Fire Daily L3 Global 1km SIN Grid V061 [Data set]. NASA EOSDIS Land Processes Distributed Active Archive Center. Accessed 2024-06-21 from https://doi.org/10.5067/MODIS/MOD14A1.061

[3] Jarvis, A., H.I. Reuter, A. Nelson, E. Guevara. 2008. Hole-filled SRTM for the globe Version 4, available from the CGIAR-CSI SRTM 90m Database: https://srtm.csi.cgiar.org.

[4] Abatzoglou J. T., Development of gridded surface meteorological data for ecological applications and modelling, International Journal of Climatology. (2012) doi:10.1002/joc.3413

[5] Abatzoglou J. T., Development of gridded surface meteorological data for ecological applications and modelling, International Journal of Climatology. (2012) doi:10.1002/joc.3413

[6] Didan, K., Barreto, A. (2018). VIIRS/NPP Vegetation Indices 16-Day L3 Global 500m SIN Grid V001 [Data set]. NASA EOSDIS Land Processes Distributed Active Archive Center. Accessed 2024-06-21 from https://doi.org/10.5067/VIIRS/VNP13A1.001

[7] Center for International Earth Science Information Network - CIESIN - Columbia University. 2018. Gridded Population of the World, Version 4 (GPWv4): Population Density, Revision 11. Palisades, NY: NASA Socioeconomic Data and Applications Center (SEDAC). https://doi.org/10.7927/H49C6VHW. Accessed 2024-06-21