A Comprehensive Guide to:
Understanding Fire Driver's Across California's Diverse Ecoregions

Primary Author(s): Erica Scaduto, Yufang Jin

For questions and comments contact escaduto@ucdavis.edu

Project Description

Wildland fires at varying intensities and frequencies are a critical ecological process across the western United States. Variability in fire behavior is heavily influenced by dynamic and often complex interactions between meteorological and biophysical components. This study uses a machine learning approach to investigate what factors are linked to the rapid spread of large wildfires in the recent decade, across six distinct ecoregions in California. Continuous daily fire spread and area burned were derived from MODIS and VIIRS active fire products. Ultimately the results from this study seek to provide insights on the efficacy of fuel management on reducing the rate of fire progression across ecologically diverse regions and help communities and managers to better anticipate and mitigate future risk of fast moving wildfires in the coming decades.

Introduction

This documentation is meant to serve as a guide to execute the central workflow of the project (Figure 1). Six primary iPython notebooks (.ipynb) files are made available under */drive/California FireTrends (2012-2020)/Scripts* to run consecutively. We recommend using Google Colab, which is a Python development environment that allows you to run Jupyter notebooks in the browser i.e. so you don't have to locally. When running the scripts using Google's Colab notebook, each step has pre-existing set-up lines to enable a smooth runtime. Although unlikely, to prevent any overloading and crashes when using Colab Pro be sure to access the high-memory runtime whenever available.

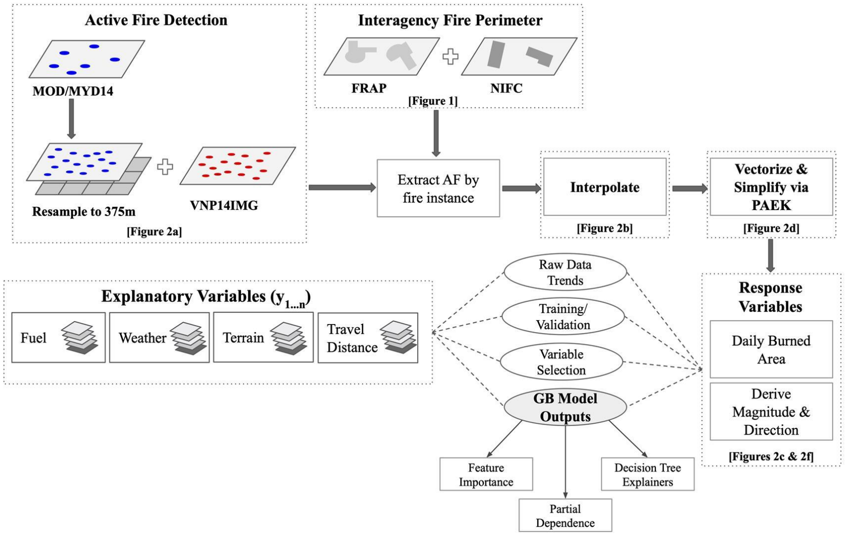

Figure 1. A comprehensive diagram outlining the general framework of the project.

Getting Started

To complete all of the necessary steps in this document, a Google account with access to Earth Engine is needed. Files and datasets can be directly accessed through the shared Drive Folder: California FireTrends (2012-2020). Please note that accessibility may be restricted to UC Davis affiliates only, additional access can be granted upon requested.

Once opened in a new browser, right mouse click (RMC) into the shared folder and add a shortcut into your own personal Drive, as seen in Figure 2. This will allow you to directly run the files and access the datasets needed in the Colab environment.

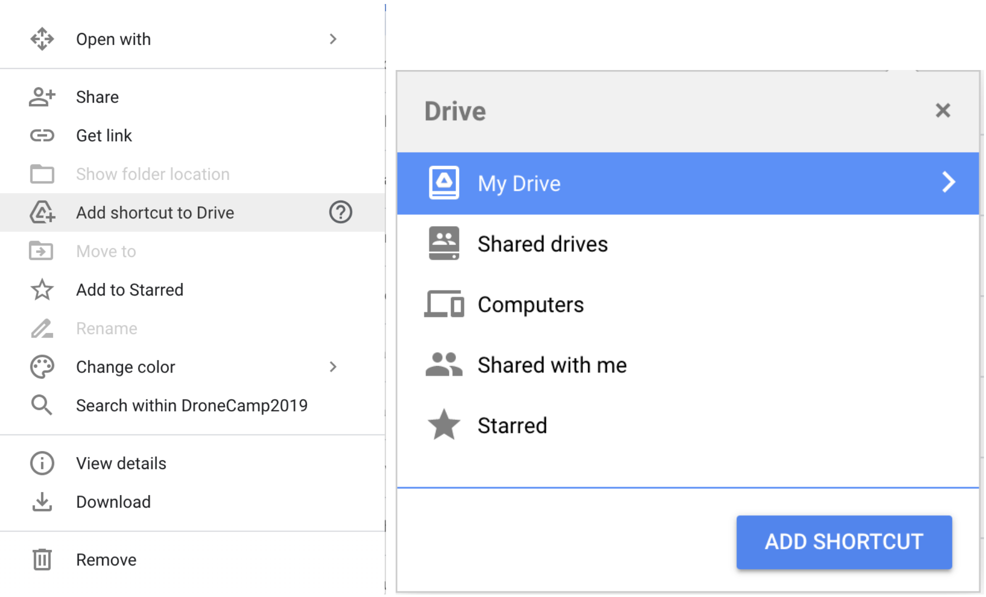

Figure 2. Screenshot of how to set up a shortcut to your My Drive folder.

Please be advised that the information and files from this project are made available solely for the purpose of documenting and sharing the primary workflows. ***When making substantial changes/edits*** *to any part of the script or dataset, please* ***make a copy of the folder*** *without editing the original files.*

Initial Steps

- Mount the colab notebook to your drive (after creating the shortcut of the shared folder):

  ```
  from google.colab import drive
  ROOT = "/content/drive"

  drive.mount(ROOT)
  ```

- Set working directory (will need to change based on step)

```
import os
rootPath  = “/content/drive/My Drive/California FireTrends (2012-2020)”
os.chdir(rootPath)
```

- Install Python packages in Colab environment, simply use

```
!pip install [package name]
```

For additional installation and usage guidelines, links provided below:

I. Setting up
   - ❏ Colab: What is Colaboratory
   - ❏ Earth Engine Python API: Python Installation | Google Earth Engine

II. Data & Spatial Processing
   - ❏ Geopandas: GeoPandas 0.8.0 — GeoPandas 0.8.0 documentation
   - ❏ Shapely: Shapely · PyPI
   - ❏ NumPy: NumPy.org
   - ❏ Rtree: Rtree · PyPI
   - ❏ GDAL: GDAL — GDAL documentation

III. Visualization
   - ❏ Matplotlib: Matplotlib: Python plotting — Matplotlib 3.3.3 documentation
   - ❏ Seaborn: seaborn: statistical data visualization — seaborn 0.11.1 documentation
   - ❏ Folium: Folium — Folium 0.12.1 documentation

IV. Modeling
   - ❏ Sklearn: scikit-learn: machine learning in Python — scikit-learn 0.24.1 documentation
   - ❏ SHAP: Welcome to the SHAP documentation — SHAP latest documentation

1. Deriving Daily Fire perimeters

Filename: [1]ActiveFires_California.ipynb

A. Interagency Fire Perimeters

Daily fire perimeters were created from active fire products based on interagency reported fire incidences. The final day perimeters were used to determine the cumulative extent of burn for each unique wildfire event. Acquisition of fire perimeters were made following the links below:

National Interagency Fire Center

- Near real-time perimeters (2020)
- Archived Wildfire Perimeters (2019-2020)
- Historic Perimeters (2000-2018)

FRAP CAL FIRE (updated annually)

- Archived Perimeters (1878-2019)

Multiple fire perimeter datasets were combined based on ‘IRWINID’, ‘GLOBALID’, in addition to the fire ‘IncidentName’ and reported year. For the purposes of this study, the perimeters were filtered to only include fires greater than 500 acres and occurring between January 2012 to September 2020.

B. Active Fire (AF) Products: Acquisition

Archive Download

- FIRMS data portal which can also be accessed via the FIRMS2SERVER

For quick access, near real-time datasets are also available for the past 7 days

- MODIS (C6)
- VIIRS (S-NPP)
- VIIRS (NOAA-20) (not used in this project)

C. Preprocessing of AF

Initial preprocessing of the active fire products include clipping to the study region i.e. state of California and converting the ACQ_DATE/TIME timezone from UTC to PST to match local data sources.

MODIS AF products were also resampled to 375m resolution representation. Individual layers for each unique day of detection were created and resampled via the ArcPy package. *Please note that due to the limitations of certain library installations in Colab, steps involving the ArcPy package (i.e. resampling, and interpolation) were done in a local windows environment and need to be run separately.* Detailed visualization can be found here.

- ❖ Saved in: *'Data/Active_Fire_Products/Filtered/MODIS_375m'*

MODIS and VIIRS AF were merged and clipped to fire extent.

- ❖ Saved based on year and fire name in *‘Data/FireInstances/***{fire year}***/***{fire name}** *’*

D. Interpolation

Natural neighbor interpolation method via arcpy library was used to derived continuous daily fire perimeters and are saved in: *‘Data/Interpolation_NonSimp’*, *‘Data/Interpolation_Interpolation’*, and *‘Data/SimplifiedSHP’* by year and fire instance. Original interpolation raster, often resulting in complex geometries were further simplified and smoothed out to reduce small island geometries and removal of non-essential vertices.

Take note that final fire perimeters are saved in Products and grouped based on fire name, and year.

E. Ignition Location

An ignition layer was created to denote incident start locations based on earliest detected hotspots from the Fire Instances shapefiles.

The DBSCAN clustering method (i.e. Density-Based Spatial Clustering of Applications with Noise) from the sklearn package was used to identify centroid locations, not limited to just one earliest datetime. This was important for fires incidents that have multiple ignition locations and to reduce error when calculating direction of spread and magnitude at daily time intervals.

- Ignition layer saved as *'Ignition/Ignition_Points/***{fire year}***/***{fire name}***'*

2. Calculating Fire Spread Magnitude and Direction

Filename: [2]MultiSpread_Vectors.ipynb

A. General Workflow

Multi-directional spread was calculated from fire-front back to previous day shared fire-line. To do this, the previously smoothed interplate daily perimeters (1D) were used to extract fire-lines for fire events greater than a day. For the purposes of this study, we only included days reaching up to 85% of cumulative burned area.

Equidistant points along firelines were created using the shapely interpolate method where it systematically generates point coordinates along a line based on a specified length. Points along the fire-front (Day_1) were traced back to closest points of the previous day shared fire line (Day_0). To account for some holes, the same method but in reverse was applied to backfill from the previous day (Day_0) to fire front (Day_1).

For spread characteristics of the first day, the ignition points from Section 1E were used, where points along the fire-front traced back to ignition location.

B. Saved File Locations

The resulting distance and direction between them were calculated and saved as *MultiSpread/CSV/***{fire name}_{fire year}***_NAT.csv*.

- Combined fire spread characteristics (i.e. each vector line for all days and all fires) are located in *MultiSpread/combined_magnitude_2012_2020.csv*
- Daily Summary Statistics which includes: maximum, median, standard deviation, as well as the lower and upper quartile values of spread distance are located in *MultiSpread/daily_magnitude_2012_2020.csv*

3. Weather Station (RAWS) Data

Filename: [3]RAWS_Extractions.ipynb

Daily weather data from Remotely Automatic Weather Station (RAWS) were acquired for fires with stations in close proximity (within 5km) through web-scraping techniques. Primary packages used include urllib3 and BeautifulSoup.

Main Url = 'https://wrcc.dri.edu/wraws/**{region}**.html'
Where {region} includes: ncalst, ccalst, and scalst

Information regarding all available weather stations including active dates, location, and station names along with their abbreviations are saved in: *data/RAWS/Raws_Info.csv*

Note: degrees were parsed and converted to decimal Lat/Long values and resaved into the csv.

Hourly raws data consolidated by fire instance can be accessed by inputting the station abbreviation {1}, day of month {2}, month {3}, and last 2 digits of the year {4} :
url = f'https://raws.dri.edu/cgi-bin/wea_daysum2.pl?stn=c**{1}**&day=**{2}**&mon=**{3}**&yea=**{4}**&unit=E'

- Saved based on fire name and year in *"data/Final_CSV/***{fire name}**_**{fire year}***_RAWS.csv"*

Hourly data were then used to calculate daily daytime averages where each day was determined to be from 6am to 6pm.

4. Gridded Layer extraction

Filename: [4]GriddLayers_Extractions.ipynb

A. Connect to GEE Python API in Colab

For this step, a GEE account is required for authentication purposes. See links in the Getting Started section for more information on connecting and using GEE's Python API in the Colab environment.

```
from google.colab import auth
auth.authenticate_user()
!earthengine authentication
import ee
ee.Initialize()
```

B. Connect to Cloud Bucket

The best way to access/import bulk datasets in Google Earth Engine is through Google Cloud bucket. Perimeters used to extract variable values were placed into a bucket which makes importing to GEE easier. These assets were used in our case as a ee.FeatureCollection.

```
! gcloud auth login
```

For the purpose of extracting environmental variables (available in GEE), we use CA's state boundary w/ 5km buffer and merged daily fire perimeters.

California 5km state boundary

- `gs://daily_fire_surfaces/CA_Extent/California_5kmbuff.shp`

Derived Fire Perimeters

- `gs://daily_fire_surfaces/ALL_2012_2020/ALL_2012_2020.shp`

C. Extracting variables from GEE

For each unique ID, based on fire name, year, and julian day, zonal statistics was performed to extract quantitative values of biophysical and meteorological conditions during the time of fire spread. All layers were resampled for consistency to 30m based on Landsat/LandFire resolutions.

I. Vegetation Indices

Using the GEE Python API, vegetation indices (i.e. NDVI, NDMI, EVI, NDWI, EVI) were calculated based on Landsat @ 30m spatial resolution at monthly and annual intervals.

Landsat 8 Surface reflectance (2013-2020)

- `'LANDSAT/LC08/C01/T1_SR'`

Landsat 7 Surface reflectance (2012-2013)

- `'LANDSAT/LE07/C01/T1_SR'`

Enhanced Vegetation Index (L8)

- `'LANDSAT/LC08/C01/T1_32DAY_EVI'`

Enhanced Vegetation Index (L7)

- `'LANDSAT/LE07/C01/T1_32DAY_EVI'`

II. Daily Weather Variables

A number of meteorological variables were obtained via Gridmet and Daymet including temperature, wind velocity, vapor pressure deficit (vpd), and relative humidity (rh). Note however, due to the lag in data ingestion for Daymet, which was only available prior to the year 2020 at the time of execution, the data source was excluded from the final analysis.

GEE band math expressions were used to calculate some additional variables. For example, the Fosberg Fire Weather Index was derived via gridmet maximum rh, maximum temperature, and wind velocity. VPD and RH were also calculated from DayMET max/min temperatures.

GridMET

- `'IDAHO_EPSCOR/GRIDMET'`

DayMET

- `'NASA/ORNL/DAYMET_V3'`

III. Topographical Variables

DEM (Elevation, Slope, Aspect)
- `'USGS/NED’`

Topographic Position Index (TPI)
- `CSP/ERGo/1_0/US/mTPI'`

Topographic Diversity
- `'CSP/ERGo/1_0/Global/ALOS_topoDiversity'`

Note: For TPI, values were regrouped to TPI zones/classes to denote ridges, valleys, and slope positions (low, middle, high). The threshold values were based on a paper noted in literature.

IV. LandFire Vegetation Layer

Vegetation Type
- `'LANDFIRE/Vegetation/EVT/v1_4_0'`

Vegetation Height
- `'LANDFIRE/Vegetation/EVT/v1_4_0'`

V. Cost Distance to Human Settlements

The Tigers 2016 roads ‘`TIGER/2016/Roads’` layer was used as a proxy for human settlement, where slope was the indicator of cost i.e. difficulty of traversing across.

Method was modified from this example:
https://developers.google.com/earth-engine/guides/image_cumulative_cost
https://gis.stackexchange.com/questions/291659/how-can-i-estimate-a-least-cost-path-in-google-earth-engine

D. Saved File Locations

Using the previously imported feature collections, we use the ee.reduceRegion to export zonal statistics of all variables for each day. The csv is saved in a local drive folder with file name based on the julian day.

Note that for certain (categorical) variables such as TPI, FBFM, EVH, EVT, etc. the percent ratio (0-100) was computed for each unique class.

Final files are located in:
- GRIDMET>> *Final_Compiled_CSV/gridmet_compiled_2012_2020.csv*
- RAWS >> *Final_Compiled_CSV/raws_compiled_2012_2020.csv*

5. Modelling Fire Spread

Filename: [5]FireSpread_Model.ipynb

A. Description

With the tabulated data sets compiled from the previous steps, we are ready to move on to the next portion which includes investigative analysis, and modelling of fire characteristics (i.e. area burned, magnitude of spread). As part of the main objective of this study, majority of analysis and modelling will group fire event days based on ecoregion.

The initial parts of this file aim to clean and pre-process the data files, consisting of renaming important explanatory variables, grouping them into categories (i.e. fuel type, fire weather, etc.), and dealing with any nonsensical data values.

Note *under the 'Read-In Data' tab we only read in either the GRIDMET i.e. primary model, or the RAWS i.e. secondary model for each run. Based on which dataset we are running, particular attention should be given to which variables are defined in a grouped variable list. However, double commented variables should be left out regardless, due to redundancy etc.*

B. Correlation Analysis

Finding linear relationships between and across variables is an important first step in quantitative analysis. In this case, the `getTop10Corr` function gives us the top ten correlations based on the absolute Pearson's r correlation coefficient values for each explanatory variable with input parameters ecoregion and dataframe.

Output displays and saves a horizontal bar plot representing the top 10 correlation values for both Magnitude (dark grey) and log(Area Burned) (light grey) variables with color coded y-axis labels based on grouped categories.

Final outputted figures are saved in:

- *Figures/Correlations/***{ecoregion}***_ALL.png*

As part of the exploratory step, regression plots color coded by ecoregion and faceted by variable is also plotted for reference and saved under *Figures/linearplot.png*. This is to mainly see if there are any obvious linear trends and/or outliers that can be detected within and across differing regions.

C. Modelling

The modelling portion of the workflow is done via the sklearn package. During the model selection process we cross-compared the performance across various machine learning regression algorithms including simple CART, Random Forest, Support Vector Machine (SVM), and Gradient Boosting. Based on the model performance metrics, mainly R^2 value, it was determined that the gradient boosting model outperformed the rest by a sufficient margin.

For the gradient boosting model parameters, while there were some tweaking and investigation involved, the optimization parameters were mainly left to default. The loss function was the least squares regression ('ls'), with a learning rate of 0.1, and an n-number of boosting stages to 1000. Values were designated based on the compromise between model performance and run time, as well as the bias-variance tradeoff.

Due to limited sample size, instead of performing the k-folds cross validation, we opted to do a simple train/testing split. The independent variables (x) were scaled using the fit transformation, standard scalar function built into part of sklearn.preprocessing. The response variables (y) were log transformed to scale outliers and normalize the distribution of the dataset.

D. Model Visualization

I. Feature Importance Plots

Permutation feature importance is especially important to identifying particular variables that are responsible for model performance. This method looks at the drop in the model's score when a single feature is randomly shuffled, indicative of how much the model depends on the feature.

Plots summarizing the overall feature importance of grouped variables were derived from the permutation_importance package under the sklearn.inspection library. Note that for these calculations, the entire dataset was used when modelling i.e. not split into training and testing sets. The scores for each variable were grouped again by variable type (i.e. Fuel Height, Topography, Weather, etc) to identify the overall percent contribution by region. These values were then plotted via stacked barplot (totalling up to 100%)

Stacked bar plots were saved in:

❖ *Figures/**{variable name}**_FeatureImportance_Stacked.png*

II. Partial Dependence Plots (PDP)

PDP gives indication of the relationship between the response and explanatory variables within the model. To create the PDP, the partial_dependence and plot_partial_dependence packages from the sklearn.inspection library were utilized.

Initial PD plots were saved in highlighting 8 pre-selected variables:

❖ *Figures/**{variable name}**_PartialDependence.png*

III. Decision Tree

There are a couple of ways to create a decision tree including the DecisionTreeRegressor package in the sklearn.tree library and the xgboost plot_tree package.

Decision trees were saved as:

❖ *Figures/Trees/**{ecoregion}**_**{res}**_tree.png*
❖ *Figures/Trees/**{ecoregion}**_**{res}**_dtreeviz.png*

Bivariate regression trees were also used to visualize the relationship between the response variable with combined effects of two variables. This was useful since fire spread typically is contingent to the combination of various physical and meteorological conditions.

IV. SHAP Explainer

To increase interpretability of the model, another method used was the SHAP value or SHaply Additive exPlantations by Lundberg and Lee (2016). SHAP values can be used to plot feature importance, partial dependence plots, and provide local interpretability. The underlying theoretical basis of SHAP is based on game theory.

Multi-dependence plots are interaction effects that aim to visualize the additional combined feature effects after accounting for the individual feature effects. Similar to why bivariate regression trees were used, multi-dependence plots were used to identify possible combined effects between certain biophysical and/or meteorologically related variables on fire spread. Most notable patterns were seen for high wind gusts and fuel type on fire activity.

Saved files can be found as:

- *Figures/Trees/***{ecoregion}_{variable A}_{variable B}***_PartialDep.png*

The ***explainer plot*** provides a local interpretation of how a model predicted the output value (in bold). The red and blue colors for any feature indicate if they push or pull a prediction higher (right) or lower (left).

Saved files can be found as:

- *Figures/Trees/***{ecoregion}***_TreeExplainer.png*

More information on shap values can be found here:
https://towardsdatascience.com/explain-your-model-with-the-shap-values-bc36aac4de3d

E. Summary Descriptive Statistics

The final portion of this section is dedicated to providing additional statistical analysis that quantitatively measures fire characteristics based on certain criteria.

Fire size and spread were grouped by lower and upper quantiles to represent small and extreme fire events. Grouped fire days were plotted against particular variables based on unique regions.

Saved figures can be found as:

- *Figures/Magnitude_Distribution_Violin.png*

Additionally, fire characteristics for the initial 48 hours of a fire event were documented to identify which regions are more prone to rapid moving fires. Similarly, days to reach 75% of cumulative burned area were also quantified, complimentary to the above metrics.

Saved figures can be found as:

- *Figures/DaysToReach75Area.png*
- *Figures/DaysToReach75Area_ByRegion.png*

6. Additional Analysis into WUI & Ignitions

Filename: [6]WUI_Ignitions.ipynb

Supplementary analysis looking into ignition occurrence within and surrounding Wildland Urban Interfaces (WUI).

For this portion of the analysis, we use Karen Short's USDA Ignition dataset (2015) saved in:

- *WUI_Ignition/Shorts_2015/Shorts_Ignition_CA.shp*

Main questions asked pertains to human influence on fire ignition from 0, 5, and 10km buffer zones. Quantification of human ignited fires that reached over 100 acres aim to inform the extent of human influence within these regions.

Notable properties/attributes to this dataset include fire name, fire size, and descriptive causes (e.g. Arson, Equipment Use, etc). Additionally, we also looked to see if there were any changes from WUI class, particularly from uninhibited to developed between the years 1990 to 2010.