Final Project Report - Andrew Nguyen

Data Overview

2.2 Data description

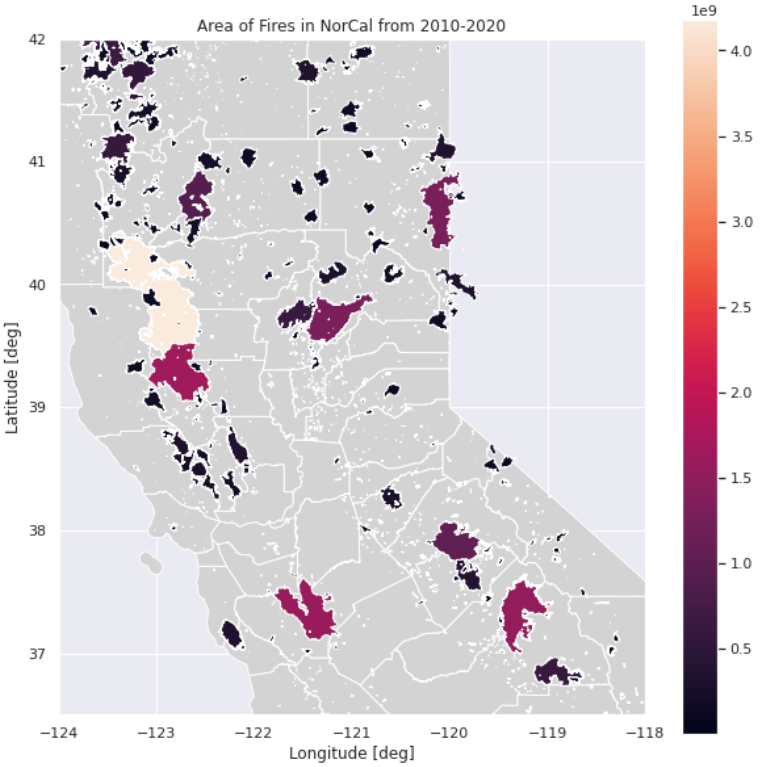
Two different datasets were acquired: one from CalFire that contained time and geographical information regarding fire perimeters up through the year 2020, and one from TerraClimate dataset that provided measurements on several wildfire-related variables such as minimum and maximum temperatures, precipitation, wind speed and soil moisture. These measurements were additionally provided on a 5 kilometer grid across the earth and aggregated to month. Naturally, geographical coordinates were also provided. The CalFire dataset initially contained 20,772 observations, while the TerraClimate dataset contained 7,338,672 observations.

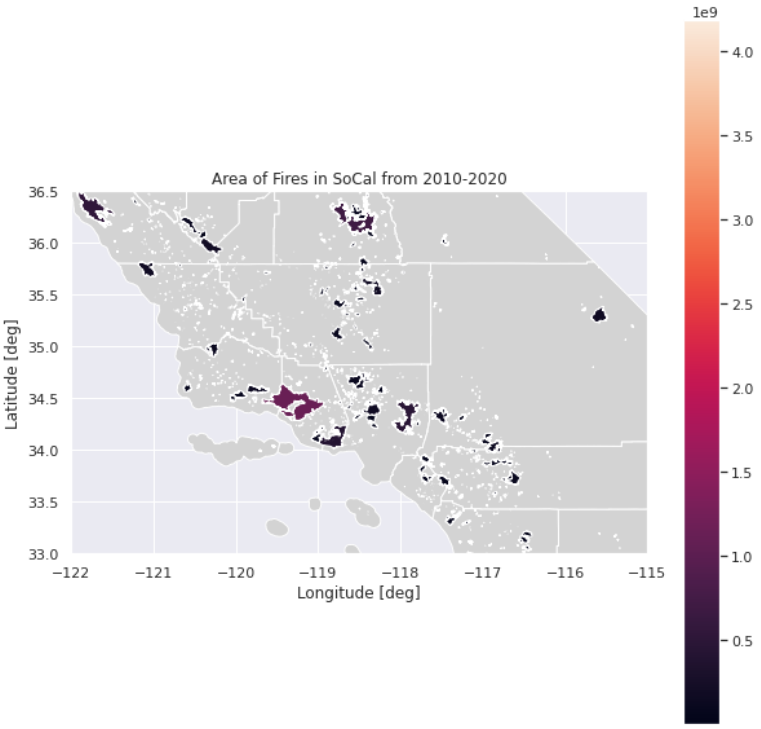
Upon closer inspection of the datasets, the CalFire dataset contained geographic data in the form of polygon objects containing coordinates that are sufficient enough to create geospatial visualizations. It contained nineteen variables in total, including the year in which the fire occurred, the state, alarm datetime, shape length and area.

The TerraClimate dataset also contained nineteen fields in total, but unlike the previous dataset, all fields were quantitative. According to the TerraClimate website, the primary climate variables included maximum temperature, minimum temperature, vapor pressure, precipitation accumulation, downward surface shortwave radiation, and wind speed. Variables derived from the ones previously mentioned are reference evapotranspiration, runoff, actual evapotranspiration, climate water deficit, soil moisture, snow water equivalent, Palmer drought severity index, vapor pressure deficit, and fire distance, which will be defined later in the report. This dataset already seemed a lot easier to work with in the modeling process than the CalFires dataset, which would mostly be used for geospatial visualizations.

2.3 Data exploration

Most of the data exploration process consisted of creating some geospatial plots and distributions of certain variables, documenting how they behave over time. For the geospatial visualizations, fire areas from 2010 to 2020 were projected onto two plots: one of northern California and southern California.

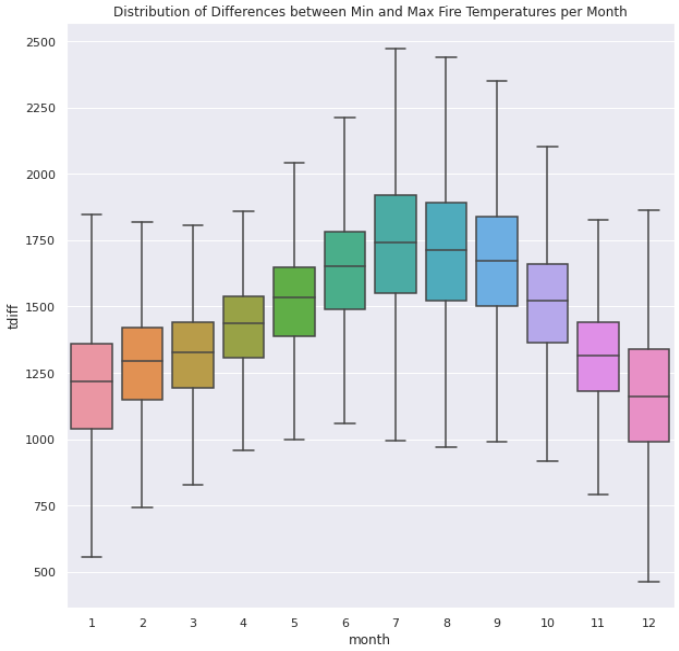


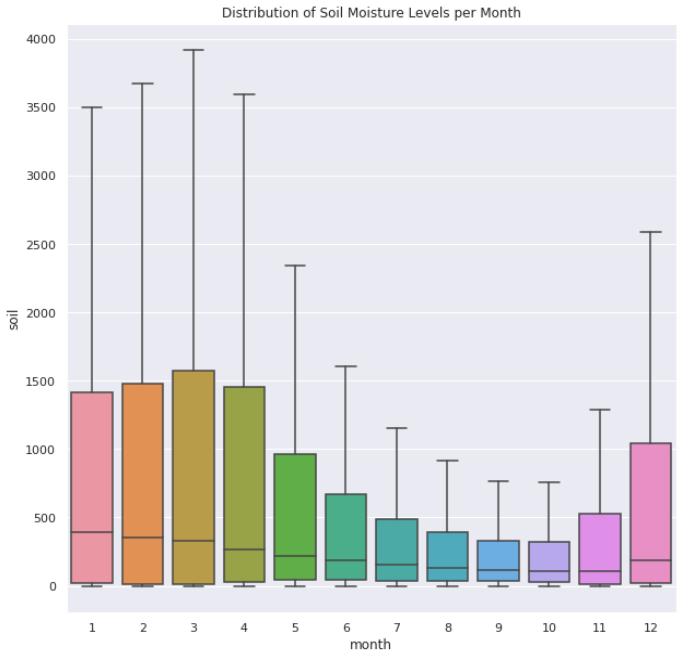


There seems to be a lot more and bigger fires in northern California than there are in southern California. One very large notable region in NorCal just above the Bay Area (specifically Santa Rosa) is most likely representing the Mendocino Complex Fires, which burned for more than

three months in 2018. The burn area was about 459,123 acres. Within the past few years, there have also been small to medium sized wildfires that burned in the Santa Barbara region as well.

Other data visualizations included differences between maximum and minimum temperatures of wildfires across months, along with soil moistures. Basically, temperature differences on average seem to be largest in the summer, while also maintaining the highest maximum temperatures. Soil moisture also typically seems to be the lowest/driest from July to October.





From these visualizations, I hypothesized that the time of year and moisture are two important variables that can decide whether there is a fire in a certain region in California. In the summer, the climate tends to be hot and arid, which can definitely contribute to the start and spread of wildfires with very high temperatures.

2.4 Data quality

The CalFire dataset looked fine to me; the variables I planned to use to make the geo visualizations looked clean and ready. On the TerraClimate dataset, there were several extremely negative max and min temperature values, along with null fire distance values.

Data Preparation

3.1 Data selection

The only manipulation I had to do with the CalFire dataset was to filter out any fires that did not take place in California, since the project only pertains to California wildfires. I simply filtered out observations whose states were not listed as "CA." A few hundred observations were removed from the process. Likewise for the second dataset, I had to figure out which observations were outside of california based on the given geographic coordinates. Using a Python package called Reverse Geocoder, I was able to search observations whose latitudes and longitudes corresponded to locations outside of California, including Nevada, Oregon and Arizona.

3.2 Data cleaning

Describe what decisions and actions were taken to address the data quality problems reported during the Verify Data Quality task of the Data Understanding phase. Transformations of the data for cleaning purposes and the possible impact on the analysis results should be considered.

Like mentioned before, there were numerous NaN values in the fire distance variable, along with negative numbers in the min and max temperature fields. Although the null fire distance values actually mean that there were no active fires during that month, I decided to still remove them anyway since I did not think they would be relevant, and the number of those nulls was negligible compared to the length of the dataset. After data selection and removal, I was left with about two million observations left, which, in my opinion, is quite plenty for modeling.

3.3 Feature engineering

Derived attributes

According to the TerraClimate website, variables derived from the ones previously mentioned are reference evapotranspiration, runoff, actual evapotranspiration, climate water deficit, soil moisture, snow water equivalent, Palmer drought severity index and vapor pressure deficit.

Generated records

Fire distance is an original variable that is calculated as the distance to the nearest active wildfire in meters. However, I used that to create a new binary variable called "fire" that would act as the response variable for the models. Based on a specified fire distance threshold, "fire" would tell me whether there was a fire in that location. I chose the threshold of 7 miles, or about 11,265 meters. On Google Maps, I searched up all fire stations' locations in California, and I estimated that on average, they were about seven miles apart from each other. I thought it would be pretty reasonable to assume that at least one fire station would be able to respond if a wildfire were to be at most seven miles away.

3.4 Data integration

No data merges were performed. Analyses were performed on the two datasets separately.

3.5 Data formatting

Min-max scaling was implemented on the explanatory variables for certain models. Only the TerraClimate dataset will be used for modeling. Before training, I decided to drop the "year" variable and keep month as the only time related explanatory variable, as I did not want to make this a longitudinal study. So, all variables were included in the training data except the year and a few other variables I manually created for data cleaning, selecting and visualizing.

Data Modeling

4.1 Modeling technique

We are trying to predict the manually created "fire" variable defined in section 3.3, which is a binary variable. So, this is a classification problem. There were four models I considered: logistic regression, support vector machine, random forest and gradient boosting classifier. For the SVM, it is typically assumed that the data is linearly separable, and with that and the logistic regression model, the data must be scaled prior to training. Besides the logistic regression, I thought an algorithmic model would be better suited for such a large dataset of over two million observations, hence the inclusion of SVM and the two ensemble methods. The ensemble methods will also be able to calculate variable importances for me to decide what factors play the biggest roles in predicting the presence of a fire.

4.2 Test design

With the threshold of the "fire" variable at a fire distance of 11,285 meters, the data became very imbalanced; the variable contained about 22 times as many zeros as ones. Thus, before splitting the data into training and testing sets, I randomly sampled 60,000 observations that had "fire" as 0 and 40,000 observations that had it as 1, without replacement. Then I proceeded to randomly split those 100,000 observations into 80% training and 20% testing sets. The model fitting/training and cross validating will be performed on the training set, while the test set is saved for last after the process is finished. Note that both subsets of data contain a six to four ratio of zeros to ones in the "fire" variable. To see how well the model performs on the far more imbalanced dataset of 22 to 1, I would predict the final model onto the rest of the data that wasn't sampled and compute the metrics from there.

4.3 Model creation

Hyperparameter settings

I only tuned the hyperparameter(s) of two of the four aforementioned models: logistic regression and the random forest classifier. For the former, I worked with the ridge regularization term, while I investigated the number of trees/estimators and maximum tree depth of the latter. I did not tune the SVM and gradient boosting models, since I ran the former at default parameters ($C = 1.0$) after the random forest classifier, and its results were not as good as the RFC. The gradient boosting algorithm was just something I learned about briefly in a previous course I took and wanted to try it out.

For ridge regression, I implemented a randomized search across a logarithmic uniform space of values between .001 to 1 to see if adding this term would reduce the effect of overfitting with many variables. Results of cross validation accuracies will be discussed in section 4.4.

After finishing the ridge regression model, I moved onto the random forest classifier. For both max tree depth and number of trees to be used, I implemented grid search across a small array of values I came up with. For max depth, I tuned across 5, 7, 10, 15, 20, 25, 30 and 40 splits. Usually by default, the nodes are expanded until all leaves are pure, or if all leaves contain less than the minimum number of observations required to split further. For the number of estimators, the default value was 100, so I tuned across that value, 150 and 200 trees.

Models produced

The final ridge regression model had the tuned hyperparameter *C* of .257, which is the inverse of the ridge regularization term, 3.89. Two random forest models were created: one with the default parameters, and one with a max tree depth of 40 and 200 estimators. The SVM was left at a default regularization term of 1, and the gradient booster was also left at its default settings.

The logistic regression model yielded these coefficients corresponding to their explanatory variables:

lat	-2.01
lon	-4.29
month	2.75
aet	-0.19
def	0.78
pet	0.68
ppt	1.84
q	-3.87
soil	-0.84
srad	-3.72
swe	0.74
tmax	2.52
tmin	1.49

vap	-3.90
ws	-2.29
vpd	-5.12
PDSI	-0.50

The random forest model gave the following variable importances:

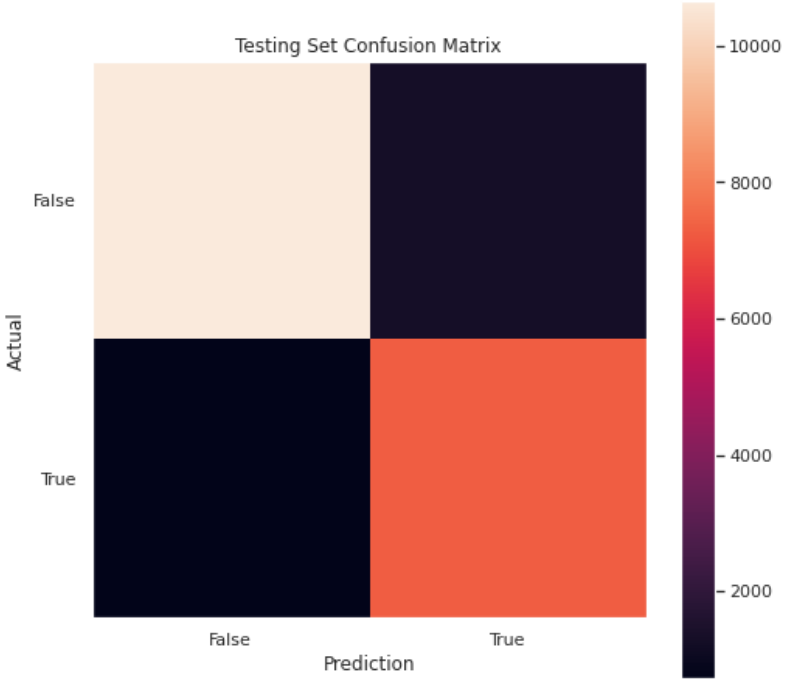
lat	.077
lon	.093
month	.067
aet	.046
def	.058
pet	.055
ppt	.043
q	.022
soil	.12
srad	.069
swe	.000044
tmax	.049
tmin	.052
vap	.057
ws	.051
vpd	.073
PDSI	.063

We can see that latitude, longitude, month, and soil moisture are the four most important variables that contribute to the presence of a wildfire in the location. Besides time of year and soil moisture as we saw in the data visualization section, it makes sense that geographic location also plays a part in predicting fires, as certain areas can have hotter and/or drier climates than others.

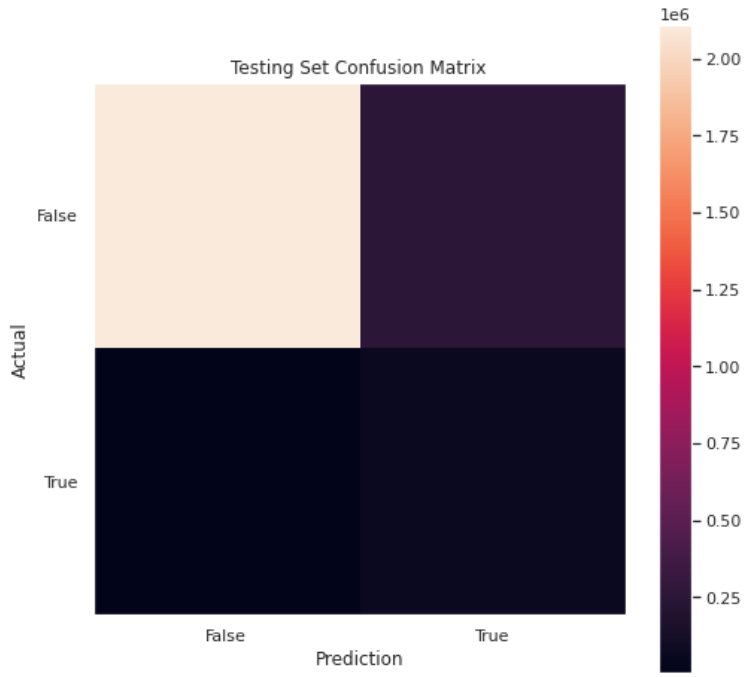
4.4 Model assessment

Ten-fold cross validation with this ridge term yielded a mean accuracy of 68.8%, which isn't great, but it was the best ridge logistic regression model at the moment. I also cross validated the model before adding the ridge term, and that yielded an average accuracy of also 68.8%; the difference was about .0004. The default random forest model gave a mean accuracy of 89.6%,while the tuned random forest gave 89.7%. The SVM gave a 10-fold CV mean accuracy of about 70% upon fitting the entire training set, and the gradient boosting gave an average of 78.1%. So from best to worst, the classification models are listed as the tuned random forest, default random forest, gradient booster, SVM, and finally both ridge logistic regressors tied together as the weakest model. Ultimately, I chose the default random forest model, as its mean accuracy seemed to have little difference compared to the tuned version, and it would be a simpler model to work with.

Upon predicting the testing data via the default random forest model, I got a testing accuracy of 89.8% and a f1-score of .877, which are pretty good metrics to see. Below is a visualization of the corresponding confusion matrix.



However, when predicting on the remaining imbalanced dataset with the same model, I get an accuracy of 89.2% but an f1-score of .33. This low f1-score was to be expected, as the model was originally trained on data that was a lot more balanced to get predictions that were as accurate as possible. From the confusion matrix below, we see that the model correctly predicted many true negatives, but a lot less true positives than the balanced testing set yielded. It also gave slightly more false positives than true positives.



Evaluation

5.1 Results evaluation

Summarize assessment results in terms of project success criteria, including a final statement regarding whether the results meet the objectives.

Overall I was able to get a pretty good classifier that was able to predict roughly 90% of the balanced dataset correctly, which in my opinion would deem a successful project. However, the model performs weaker on very imbalanced datasets, which typically reflect the real world better than balanced datasets do in the context of this study. It is great that I was able to predict many true negatives on the imbalance dataset, but with fewer true positives it may raise issues when it comes to the alert of fires in the area. I would say the project is mostly a success, as the model predicts wildfires well on balanced data and technically the imbalanced one purely based on accuracy, but when looking at other metrics, the latter will give weaker results.

5.2 Discussion

The data mining engagement was quite good for the random forest model; it was able to yield the highest training accuracies out of all the models tested. If more time or better time management were allotted, some hyperparameter tuning on the SVM and gradient boosting models would

have been insightful as well. Implementing manual or formal/algorithmic variable selection for the logistic regression and perhaps the SVM model could have also aided in the process. Also, since it appears that the ensemble methods performed the best, in the future, more research on those types of models can be done to investigate if any different ones can give better scores, especially on imbalanced data.

Like mentioned before, the model performed well on both testing and imbalance data in terms of accuracy. Because of the imbalance however, there were more false positives than true positives when predicting on the remaining unsampled observations. In the context of this study, there are more false alarms regarding the presence of wildfires, which could unnecessarily force fire stations into preparing to fight a fire when there really isn't in their vicinity. On the bright side, there are fewer false negatives, which in my opinion leads to more severe consequences: firemen not responding to a fire when there really is one. Although there are environmental benefits to wildfires, it can easily cause more damage if left uncontrolled. Personally, I would be hesitant to deploy the model, but if it ultimately were to, I would not rely on it too much and make use of human and field knowledge alongside it.