Can we predict fire with weather data?

Harsh (hp444), Yi Yao (yy899), Murali (mt788)

November 7, 2019

Abstract

In this report, we explore the idea od using the weather parameters such as temperature, pressure, etc to predict if a wild fire may occur. The datasets in use are the *Wild Fires Dataset* and the *Weather Dataset*. The datasets were combined to use realistic weather features as inputs along with binary *fire/no fire* output. This is clearly a binary classification problem and a very imbalanced one. Dealing with the imbalance is the highlight of this report. We use precision and recall metrics to form the dataset. We attempted to use algorithms like *SVM, XGBoost* and also autoML (*TPOT*). We show our preliminary results on how we deal with imbalance and also how we plan to work on this issue in the next half of the semester.

1 Introduction

The weather has literally been the **hot topic** of discussion in recent times. One aspect of weather data is that, when it is combined with other data sets, it can be used to predict a lot of different aspects related to society. For example, one interesting aspect could be how does the traffic condition changes based on the weather in a particular city or if there is a correlation between crime rates and weather conditions. In this project, we are trying to address a more important issue of predicting wildfires in different cities based on weather conditions. Since the future weather data is readily available these days, a good model would help predict these wildfires in advance and take necessary precautions to completely avoid it. This would help avoid human/wildlife loss.

2 Data Statistics

Here is a quick look into the data that we're working with. Table 1 shows the statistics of the weather data and Table 2 shows the statistics of the fire data.

Table 1: Weather Data Statistics

Elements	#
Cities	36
Features/city	9
Total Samples	1.63 million
Samples/city	45 thousand

Table 2: Wild Fire Data Statistics

Elements	#
Cities	36
Features/city	9
Total Samples	1.63 million

3 Data Cleaning/ Data Exploration/ Data Viz Website

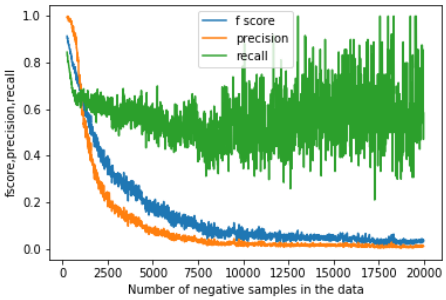
First, we combined both the datasets to arrive at one single table containing the weather features and simply attributing fire/no fire column. This was not straight forward because the fire data had latitude and

longitude for refernce while weather data had cities. The first challenge was to identify the slack to allocate cities to the co-ordinates. We gave a 1° (which is roughly 100 km) which was a safe distance to associate with the city weather data. After this, we had to parse the tables with weather features and the binary fire/no fire feature. This resulted in 10 features in X space and one feature in y space. We had to perform one-hot encoding for one of the parameters which was weather description because of which we were left with 33 features at the end.

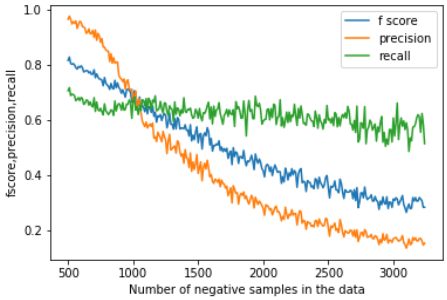
In San Fransisco, the number of fires in the set was 1260 and the number of no-fires is roughly 43 thousand. This would cause issues with overfitting. Using the precision and recall metrics, we actually find that the number of data points where the tradeoff occurs is with 1400 negative examples with 1400 positive samples or in some sense almost equal number of positive and negative examples. Figure 1(a) and Figure 1(b) show the number of negative examples needed to ensure tradeoff between precision and recall. Also, we know that California is burning at the moment, Figure 1(d) shows the distribution of number of fires in the SF city grouped by hour where fires where actually reported. If you want to look at fires in the cities we are covering, Figure 1(d) shows a screen shot of one instant of time. The animation showing fires over time is hosted on *Github. Follow this link and please allow for sometime for the website to load.*

Figure 1

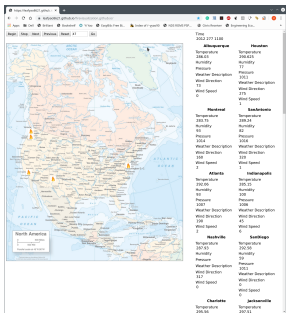
(a)



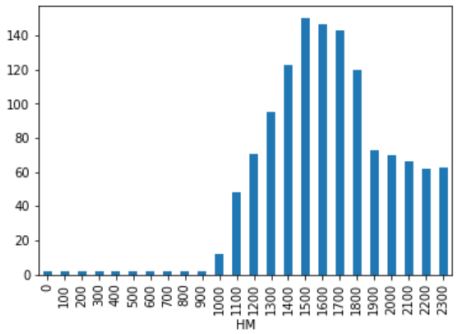
(b)



(c)



(d)



4 Data Analysis

We decided to use two ensemble learning models, namely XGBoost and Random Forest, as well as a linear model, namely Soft-Margin SVM with the RBF (Radial Basis Function) kernel. On top of this, we also ventured using automl framework TPOT. The following table summarises the results for two cities.

XGBoost works the best because of the following reasons:

Table 3: SanFrancisco

Classifier	Meta Parameter	Training Accuracy	Test Accuracy
	n_estimators		
XGBClassifier	63	0.6677	0.6587
RandomForestClassifier	85	0.6979	0.6865
	C		
SVC	0.5000	0.5099	0.5417

Table 4: Atlanta

Classifier	Meta Parameter	Training Accuracy	Test Accuracy
	n_estimators		
XGBClassifier	75	0.7103	0.7098
RandomForestClassifier	91	0.7297	0.7319
	C		
SVC	0.5000	0.6626	0.6788

5 Work for the upcoming term

5.1 Bootstrapping

To avoid overfitting problem, we may try out the following pseudocode hoping to solve the problem:

Algorithm 1 Avoid Overfit using our version of Bootstrap

DPos ← Positive data and DNeg ← Negative data
for Number of Iterations **do**
  Sample negative data in equal proportion to available positive data
  Fit the data and store the weights
end for
Average the weights to get an averaged model
Test this model on a new and unseen data for validation
Require: Changing the ratio of sampling in the dataset to average different models
Test this model on a new and unseen data for validation

5.2 Feature Engineering

Feature Engineering plays a key role in cases like ours where our data is very restrictive within certain ranges. We plan to use different feature transformations like exponentials, logarithms. Apart from this, we may also explore into using different data available online for more feature information related to fires and