Wildfire Burn Area Prediction

Adam Stanford-Moore
sasm@stanford.edu

Ben Moore
benmoore@stanford.edu

Abstract

Over the past few decades, wildfires have been steadily increasing in number and severity, with over 52,000 fires in 2018 alone [1]. We use several different models to predict wildfire burn area, based on data from 1.88 million historic fires from throughout the United States and during the period of 1992-2015. We investigate using linear regression, a neural network, and a support vector machine (SVM), and evaluate our models' performances with mean absolute error (MAE) and mean absolute percent error (MAPE). Though challenged by a huge imbalance in fire sizes, our best model, an SVM, was able to predict fire size on our test set with mean absolute percent error of 41%.

1 Introduction

1.1 Motivation

In recent years wildfires have destroyed entire communities and damaged billions of dollars worth of property. The unpredictability of fire sizes makes it especially difficult for fire departments to effectively allocate the appropriate resources to mitigate damage, so a reliable model predicting forest fire burn size is a necessity for disaster management. We have developed a model that uses historical fire data and some weather parameters to predict how large a fire would grow from numerical starting parameters. The input to our algorithm are numerical features such as date, latitude, longitude, and the output is the final fire area in hectares.

1.2 Related Work

Radke et al (2019) built a convolutional neural network (CNN) to predict the area the current fire is expected to burn in the next 24 hours [2]. The CNN was trained on location specific input which was heavily restricted by the small size of the dataset, so as a result they used a number of data augmentation techniques.

Tehrany (2019) produced a fire susceptibility heatmap of areas in Vietnam using historical fire data, climatic and topographic data, and distance to residential areas [3]. The study ultimately used a LogitBoost ensemble-based decision tree (LEDT) model but also applied support vector machine (SVM), random forest (RF),and Kernel logistic regression (KLR) models.

Castelli et al. (2015) used geographical and meteorological conditions to predict fire burn area [4]. The meteorological data included traditional climatic data such as humidity, wind speed and temperature and indexes such as fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), and Initial Spread Index (ISI). In this study a number of machine learning techniques were applied and compared including Naïve Bias, Decision Trees, Support Vector Machine, and Random forest.

Research groups from previous iterations of CS229 have also made contributions in this area. [5] used a collection of geospatial data from a spectroradiometer aboard NASA's Terra/Aqua satellites and land-based weather stations to predict the fire-risk at a particular location based on the meteorological conditions at that location up to three months prior. All three methods explored logistic regression, boosted trees, and multilayer perceptron and yielded similar results with about 75% accuracy in fire prediction. [6] predicted the area consumed by fires in Montesinho natural park in Portugal based on weather and fire data collected there over three years from Jan 2000 to Dec 2003, but experienced a lot of difficulty.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2 Datasets

We predict fire sizes on two main datasets: the Kaggle Wildfire dataset and the UCI dataset.

2.1 Kaggle Wildfire Dataset

The Kaggle dataset is a collection of 1.88 million wildfires that occurred throughout the entire US during the period 1992-2015. Each fire is reported with its burn area in hectares and many characterizing codes (no weather features), of which we chose 6 numerical ones: the year, date fire started (continuous date), discovery day of the year (1-365), fire cause code (1-13) specifying what started the fire, latitude, and longitude. The distribution of fire sizes is given in Figure 1, where its clear that although fire sizes range from 0.0001ha (1 square meter) to 600,000 ha (2300 mi^2), the majority (by several orders of magnitude) of fires in the dataset are about 10 ha in size.

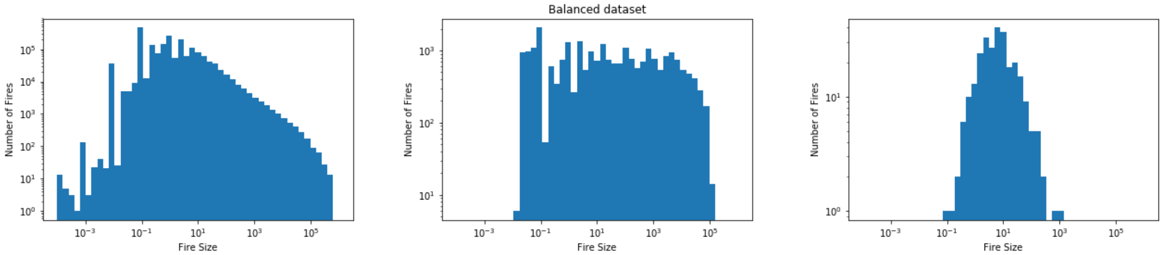

Figure 1: (Left) The Kaggle fire sizes distribution (log-log scale) indicates that fires are historically very small with fewer major ones. (Middle) Kaggle dataset after balancing through randomly removing fires with over-represented sizes. (Right) The UCI dataset has fewer fires, and a narrower distribution of sizes.

As a result of the large imbalance in fire sizes, we also trained on a version of our Kaggle dataset in which we balanced the fire sizes more evenly by undersampling: randomly removing over represented fire sizes. To do this, we split the fire sizes into order of magnitude bins and randomly sampled 4000 fires from each bin. Although the balanced distribution is more flat, as in Figure 1, it has significantly fewer examples, only 33,000 in the training set instead of $>$ 1.7 million fires. We continued to test on the original distribution of fires.

2.2 UCI Dataset

Our second dataset, from the UCI machine learning repository is a collection of 512 fires from a large national park in northern Portugal. Unlike the Kaggle dataset, this one provides weather features associated with each fire. We used most of the provided features, which included an x and y location within the national park, month, day of the week, Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), temperature (Celsius), relative humidity, wind, and rain, to predict the burn area.

We randomly split the massive Kaggle dataset into 96% train, 2% validation and 2% test and split the smaller UCI dataset into 60% train, 20% val and 20% test.

2.3 Feature Analysis

Based on our literature review we assumed that some features such as wind, temperature and humidity would be highly correlated with fire size, but given an initial exploration of the data it was clear that even these highest correlated features of the dataset were still only weakly correlated. As shown in figure 2 wind has little to no correlation to fire size in the UCI dataset. Figure 2 also shows the concentration of massive fires around summer time compared to more uniform distribution of smaller fires throughout the year.

To quantify the correlation between the dataset features we produced correlation matrices for both datasets. From figure 3 we can see that all the features in UCI dataset are poor indicators of fire size with the highest correlation magnitude, 0.076 attributed to relative humidity (RH). The correlation matrix of the Kaggle dataset has fewer features and no weather features. All mutual features of both datasets have smaller correlation magnitudes in the Kaggle dataset compared to the UCI dataset. This makes it very difficult to predict fire size well and also demonstrates the importance of weather features to predict fire size which have a relatively high correlation to fire size compared to the Kaggle dataset.

We applied Principal Component Analysis (PCA) to further understand which features explain the variance in the datasets and which are redundant. Figure 4 confirmed that weather data is essential as the weather features are heavily

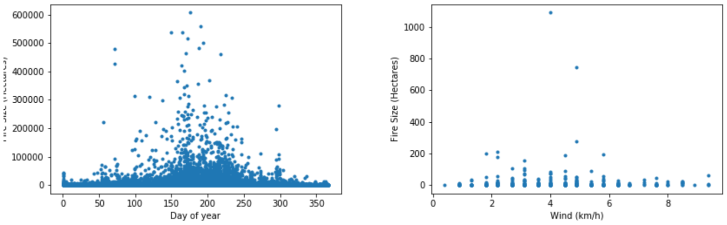

Figure 2: (Left) Most of the really large fires in the Kaggle data happen during the peak of summer, but small fires are well distributed. (Right) Windspeed (km/h) is not well correlated with Fire size.

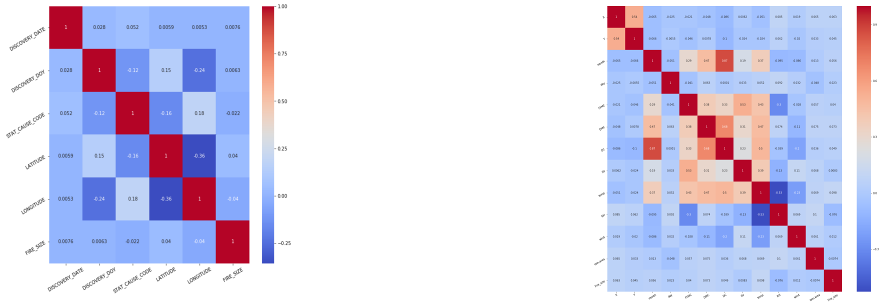

Figure 3: Feature Correlations with Fire Size. (Left) Kaggle dataset, (Right) UCI dataset

weighted in the first component representing the UCI dataset. Interestingly relative humidity and wind do not represent as much of the variance in the data as initially thought and are only included with a large weighting in the third and fourth components respectively.

3 Models

3.1 Baseline Linear Regression

As an initial baseline model we applied a linear regression model to both datasets.

3.2 SVM

We also implemented a number of SVMs by experimenting with weighted class balancing and comparing a number of various kernels and orders of polynomials; such as linear, sigmoid, polynomial and radial basis function. Of the kernels, the radial basis function was the best predictor.

3.3 Neural Network

We next designed and tuned two separate neural networks for each dataset. For the Kaggle dataset we designed a simple 10 layer network to predict with the same parameters and train/test split. The layers were fully connected with between 32-256 nodes with dropout of 0.2 on each node and Relu activation on all nodes except the output node which has linear activation. The model has $\sim 150,000$ trainable parameters and is significantly more flexible than the linear regression baseline. We used an Adam optimizer, trained over 10 epochs, and based our loss function on the mean-absolute-percent-error (which worked better than mean absolute error or mean squared error due to the large variance in fire sizes).

For the UCI dataset we designed a seven layer neural network, consisting of six hidden layers with 1500 neurons in each layer. We tuned the number of hidden layers used and the number of neurons in each layer using a simple grid search. Again we used the Adam optimizer, but instead opted to use a squared hinge loss as the network was not learning using the MAE score.

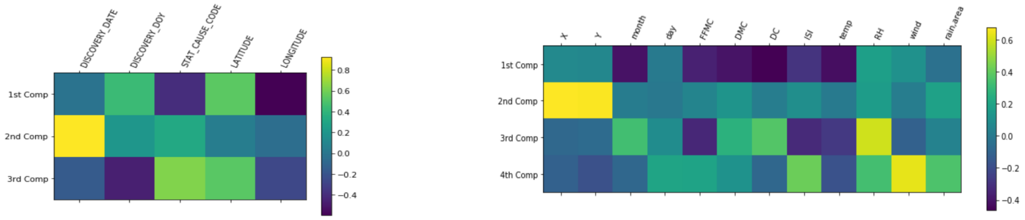

Figure 4: (Left) PCA - Kaggle dataset: variance explained mostly by longitude and discovery date. (Right) PCA - UCI dataset: variance explained mostly by the weather features and location.

3.4 K-nearest-neighbors

We implemented the K-nearest-neighbors algorithm, which stores all available train data and classifies test data based on a similarity measure. The algorithm uses a set number of training samples closest/similar to a test point by some specified distance (Euclidean/Manhattan/Minkowski) and predicts the label of this point based on these neighbors [7]. We applied a number of different distance metrics to this problem: Euclidean, Manhattan and Minkowski. We also experimented with uniform and distance weighted weights applied to the neighbors of a point, of which the uniform weights performed best.

3.5 Decision Tree

We also implemented a Decision tree, which learns simple decision rules from features of the training data to predict the value of a target variable [8]. The tree algorithm also enables an insight into the features that the model learns to be most important in predicting fire size, via univariate feature selection. Univariate feature selection examines each feature individually to determine the strength of the relationship of the feature with the response variable.

3.6 Stacked Regressors

As we saw a number of models perform well on different fire sizes we applied stacking, which is an ensemble method used to combine multiple estimators to reduce their biases. The predictions of each individual estimator are input to a final estimator to generate the final prediction [9].

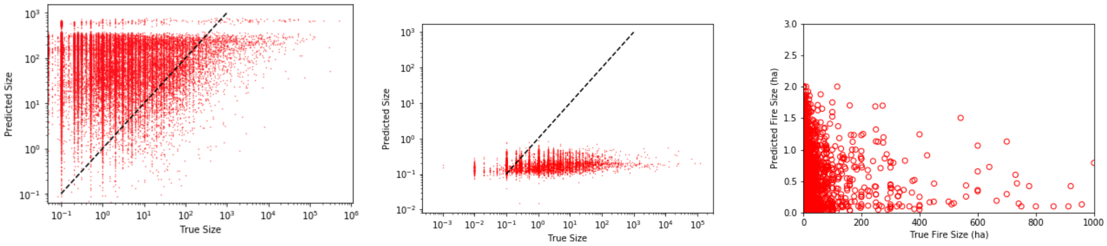

Figure 5: Comparison of predicted fire size (vertical axis) vs true size (horizontal axis) for linear regression (left) SVM model (middle) and k-nearest neighbors (right), all on the Kaggle dataset. The black dashed line would be perfect agreement.

4 Results/Discussion

We trained our models on the three datasets – Kaggle, balanced Kaggle, and UCI – and evaluated our models on the respective test sets based on two metrics: mean absolute error (MAE) and mean absolute percent error (MAPE). Table 1 summarizes the performances of the models.

The linear regression model performs surprisingly well on the UCI dataset given its simplicity, as seen in Figure 5. The model tends to overpredict fire sizes yet with an MAE of 171 hectare (ha) and MAPE of 82500% on the Kaggle dataset, establishes the baseline score to beat.

The SVM had the overall best prediction ability of all models across all three datasets for both MAE and MAPE metrics. With MAPE of 41% on Kaggle balanced and 78% on UCI, on average the model outputs a reasonable estimate of fire size. To achieve this performance we tuned the kernel type and in the case of a polynomial kernel we tuned the degree of the polynomial to fit. The kernel which achieved the lowest MAE and MAPE scores, 6.334 and 77.711 respectively, was the radial basis function kernel. As in Figure 5, the variance of predictions is much smaller than the variance of true fires, even for the balanced Kaggle dataset.

In the training curves of our neural network (Figure 6) trained on the Kaggle dataset, we see that our training and validation loss defined by MAPE, do not significantly decrease with more training, even with ample flexibility in free parameters. This perplexed us, but perhaps it confirms that the parameters that we have access to do not fully capture and accurately predict the fire sizes, i.e. the correlations of our features to fire sizes are small. Performance on the balanced dataset was better, and achieved more significant learning. On the UCI data, the network performed still better on MAE (8.26 ha), likely because the range of data was smaller.

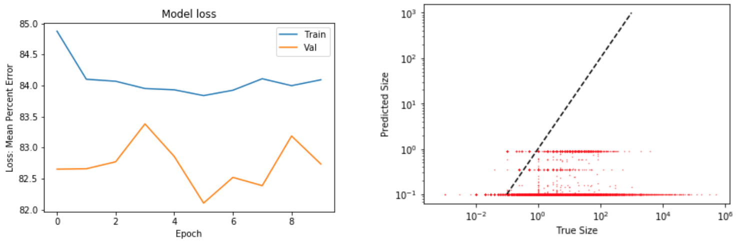

Figure 6: (Left) Training was challenging as the train/validation losses did not significantly improve over time. (Right) The neural network largely predicted smaller fire sizes, which greatly improved the MAPE score over linear regression, but totally missed the fewer large fires.

Some of the best models, like the SVM and neural net repeatedly predict only a small range of fire sizes in order to minimize the MAE and MAPE scores. In comparison the linear regression model appears to have a larger variance in its predictions (Figure 5), but does not achieve MAE and MAPE scores comparable to the neural network or SVM. Balancing the Kaggle dataset produces systematically better scores and allows models to output predictions with more variance, but the low variance predictions are still observed.

Table 1: Model Test-set performances

	Kaggle		**Kaggle-Balanced**		**UCI**	
	MAE	MAPE	MAE	MAPE	MAE	MAPE
Linear Regression	171.12	82531.18	164.9	79895	15.547	869.165
SVM	52.25	50.91	35.57	41.0	6.334	77.711
Neural Network	87.29	82.51	36.12	43.3	8.264	347.628
K-nearest-neighbors	111.75	3190.5	36.91	43.39	15.53	639.88
Decision Tree	118.19	4119.8	36.9	42.9	31.5	995.8
Stacked Regressors	53	1326.9	36.92	43.39	9.45	64.54

5 Conclusions / Future Work

In conclusion, several of our models achieved relatively high scores on our evaluation metrics including SVM, which had the best performance overall with a MAPE of 41% on the balanced Kaggle set. This is the model we would choose to help firefighters predict how large a fire will grow to be.

A large challenge throughout the project has been negotiating the imbalanced distribution of fire sizes, in which there are several orders of magnitude more fires under 10 ha, than larger. To help reduce the class imbalance, we undersampled, though there are other methods like synthetic sampling, augmenting, and oversampling. These additional balancing strategies would be interesting to investigate.

Given that the neural network overfit to the UCI training data (indicating a high variance issue), increasing the UCI dataset size might improve performance. We found that the Kaggle dataset was extensive, but had fewer relevant features, while on the other hand the UCI dataset had nice features, but was smaller and had a narrower range of fire sizes. Additionally, transforming the problem to a classification problem is another option but we suspect it might only return similar results in a different format given the lack of high quality and volume data. We would love to be able to combine more weather and environmental factors, and link in geospatial satellite imagery of terrain to help in prediction.

6 Contributions

Ben and Adam worked well together meeting frequently to brainstorm ideas and come up with to-do items that they did on their own. Ben layed a lot of the foundational code cleaning/reading the data and setting up the structure for the neural network. Adam fleshed out and experimented with the neural network and did a lot of feature and error analysis. They worked closely together following the milestone in fleshing out the models, writing up this paper, and making the poster.

See our code at `https://github.com/benkmoore/Wildfires/tree/Ben` and `https://github.com/adamstanfordmoore/Wildfires`.

References

[1] Insurance information institute - wildland fires. `https://www.iii.org/fact-statistic/facts-statistics-wildfires`.

[2] David Radke, Anna Hessler, and Dan Ellsworth. Firecast: Leveraging deep learning to predict wildfire spread. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4575–4581. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[3] Mahyat Shafapour Tehrany, Simon Jones, Farzin Shabani, Francisco Martínez-Álvarez, and Dieu Tien Bui. A novel ensemble modeling approach for the spatial prediction of tropical forest fire susceptibility using logitboost machine learning classifier and multi-source geospatial data. *Theoretical and Applied Climatology*, 137(1):637–653, Jul 2019.

[4] Mauro Castelli, Leonardo Vanneschi, and Aleš Popovič. Predicting burned areas of forest fires: An artificial intelligence approach. *Fire Ecology*, 11:106–118, 04 2015.

[5] Caroline Famiglietti, Natan Holtzman, and Jake Campolo. Climate hazards group infrared precipitation with station data. *CS229*, Dec 2019.

[6] Eliza Klyze, Sun Cuthbert, and Otomo Rysen. Wildfire burn area prediction. *CS229*, Dec 2017.

[7] J. M. Keller, M. R. Gray, and J. A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(4):580–585, July 1985.

[8] Yoav Freund and Llew Mason. The alternating decision tree learning algorithm. In *icml*, volume 99, pages 124–133, 1999.

[9] Harris Drucker. Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115, 1997.