MACHINE LEARNING ENGINEER CAPSTONE PROJECT

Mariolys Rivas.
February 13, 2021.

I. DEFINITION

Project Overview

Wildfires are a major issue in the United States and Canada, and are particularly damaging to humans and nature in the West Coast, where I reside. Once a regular and sometimes beneficial natural phenomenon, wildfires have become more destructive and disruptive in the past decade.

Climate change continues to be the most important environmental issue of our generation. The global average temperature of Earth has continued to grow rapidly over the past few years, accelerating the onset of drier and hotter conditions that help make natural disasters such as wildfires and flooding more frequent and catastrophic than ever before.

Determining the causes of a wildfire, such as burning debris, lightnings, or campfires, can help governments tackle those causing factors in the future. Predictive machine learning models can help us guess this cause with certain confidence. In this project I aim to build one such model, while illustrating core components of the machine learning engineering process, including data exploration, data processing, model training, refinement, and deployment.

Kaggle's 1.88 Million US Wildfires dataset contains 24 years of United States geographic and temporal wildfire data, and will be my investigation's data source.

Problem Statement

My goal is to build a machine learning model inspired by the following question posed under the "Inspiration" header of Kaggle's 1.88 Million US Wildfires dataset;

"*Given the size, location and date, can you predict the cause of a wildfire?*"

More specifically, my model's input will include the **wildfire's date**, **time of discovery**, [geopolitical] **state**, and **estimates size** (in acres). Its output will be a **cause** amongst all possible causes in the dataset. More details into these fields can be found in the subsequent sections.

Following standard machine learning conventions, the project design to produce the expected predictive model will be as follows:

- Divide the data into three different sets: training (~80%), validation (~10%) and testing (~10%).
- Data exploration and preprocessing, including visualizing the existing data, completing missing values (if any), and other pre-processing strategy to get the data ready for training.
- A **benchmark** model will be defined as always picking the most frequent wildfire cause.
- Model training under three different methods: **XGboost**, **Linear Model**, and **MLP** (neural network).
- These models are to be compared (among each other and with the benchmark) according to the metrics defined in the next section.

Notice that if there are N possible wildfire causes in the dataset, then the benchmark model above should produce correct predictions for at least $1/N$-th of the samples. I expect at least one of the trained models to perform better than this.

Metrics

I plan to use both the **accuracy** of each model, and their F_1 scores as metrics. For each possible output label (in this case a wildfire's cause), the F_1 scores provide a good balance between **precision** (the fraction of this label's predictions which are accurate) and **recall** (the fraction of wildfires produced by this cause which were accurately predicted). This balance is especially important if the causes are imbalanced, meaning that some of them are a lot more frequent than others.

The F_1 score is defined as follows:

$$F_1 = 2\frac{Pr \times Re}{Pr + Re}$$

Where Pr represents precision and Re represents recall. Notice again that this measure provides a single value for each output label. These values can be averaged over all labels in order to compare two different models.

II. ANALYSIS

Data Exploration

As mentioned above, I am using **Kaggle's 1.88 Million US Wildfires dataset**, containing 24 years worth of US wildfires data (1.88 million data points). The original dataset consists of more than 40 feature columns, which I narrowed down to the 10 columns below:

	FIRE_YEAR	DISCOVERY_DOY	DISCOVERY_TIME	LATITUDE	LONGITUDE	FIRE_SIZE	FIRE_SIZE_CLASS	STAT_CAUSE_CODE	STAT_CAUSE_DESCR	STATE
0	2005	33	1300	40.036944	-121.005833	0.10	A	9.0	Miscellaneous	CA
1	2004	133	0845	38.933056	-120.404444	0.25	A	1.0	Lightning	CA
2	2004	152	1921	38.984167	-120.735556	0.10	A	5.0	Debris Burning	CA
3	2004	180	1600	38.559167	-119.913333	0.10	A	1.0	Lightning	CA
4	2004	180	1600	38.559167	-119.933056	0.10	A	1.0	Lightning	CA

Although future work can be done using every one of these features for prediction, I have decided to simplify this further to the following input and output variables:

FIRE_YEAR = Year the wildfire happened. **1992 - 2015**.
DISCOVERY_DOY = Day of year on which the fire was discovered or confirmed to exist. **1 - 366**.
DISCOVERY_TIME = Time of day that the fire was discovered or confirmed to exist. **0000 - 2359**.
FIRE_SIZE = Estimate of acres within the final perimeter of the fire. **0 - 606,945.0**.
STATE = Geopolitical state (CA, TX, etc).

Remark: It is possible to realize this investigation using **latitude** and **longitude** instead of just the state, perhaps achieving better accuracy. However the state by state breakdown has the interesting side effect of making it possible to assess relevant differences between state distributions by studying feature importance. This is done briefly in the **Free-Form Visualization** section.

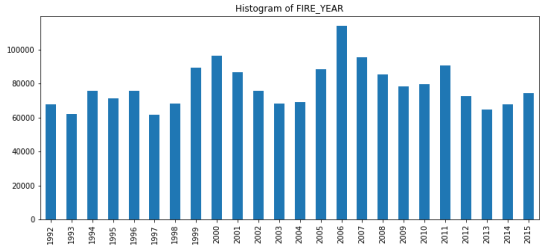
The target (output) variable is the following:
STAT_CAUSE_CODE, STAT_CAUSE_DESCR = Code and description for the (statistical) cause of the fire. Below is the distribution of these values in the dataset:

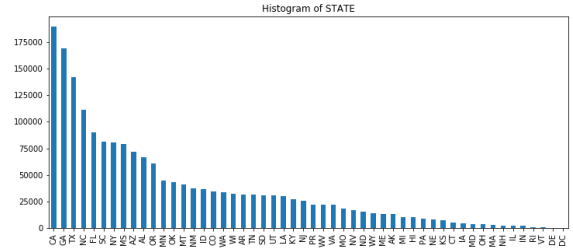
STAT_CAUSE_DESCR	STAT_CAUSE_CODE	FREQUENCY
Arson	7.0	281455
Campfire	4.0	76139
Children	8.0	61167

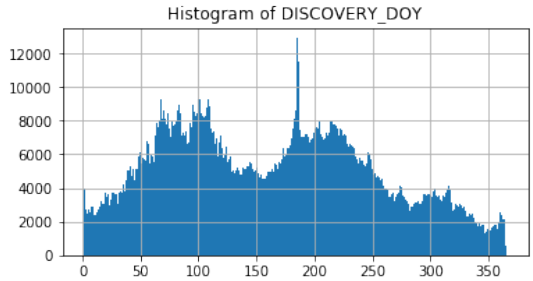
Debris Burning	5.0	429028
Equipment Use	2.0	147612
Fireworks	10.0	11500
Lightning	1.0	278468
Miscellaneous	9.0	323805
Missing/Undefined	13.0	166723
Powerline	11.0	14448
Railroad	6.0	33455
Smoking	3.0	52869
Structure	12.0	3796

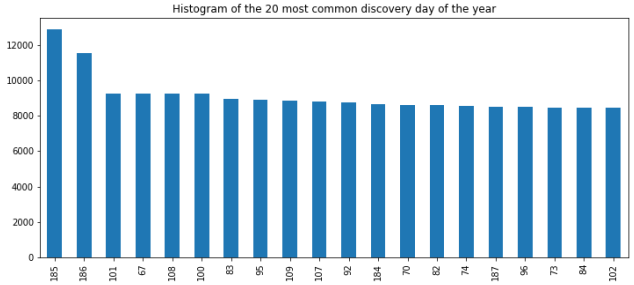
Exploratory Visualization

It is possible to get a preliminary sense of the input values in the dataset by independently looking at the histograms of some of the features



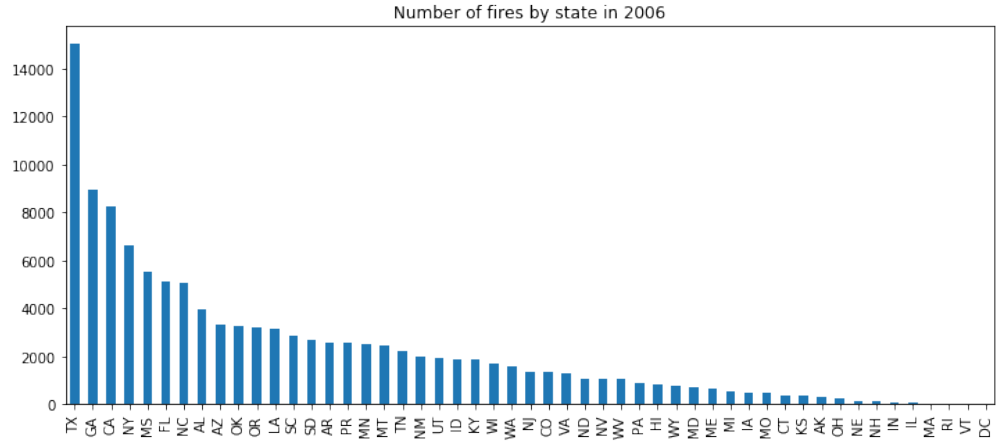




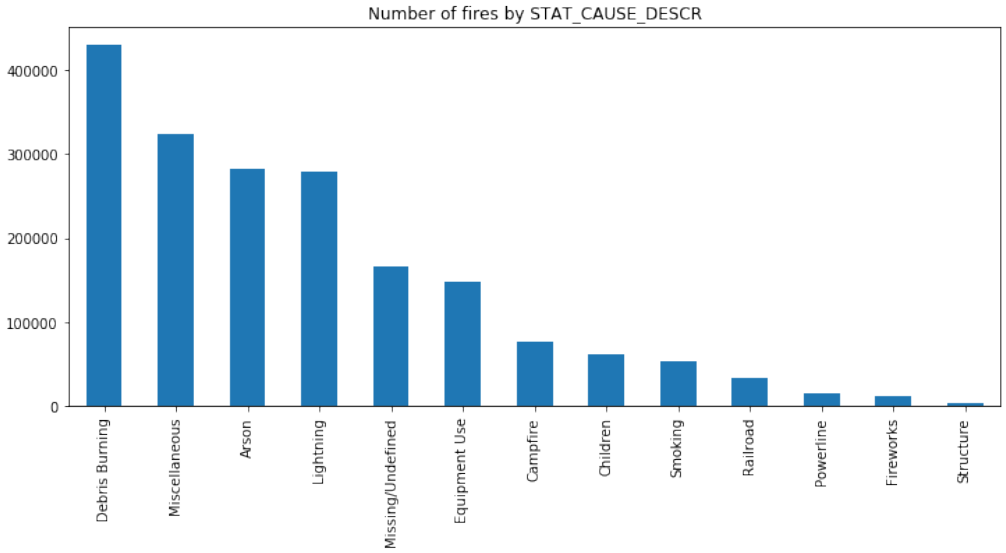


An interesting observation is that the day 185 of a year (186 for leap years) is July 4, which suggests that fireworks are very likely to cause fires during July 4 celebrations.

Also 2006 is the year with the most wildfires in the dataset, which happens to be a year when Texas had a major wildfire event:

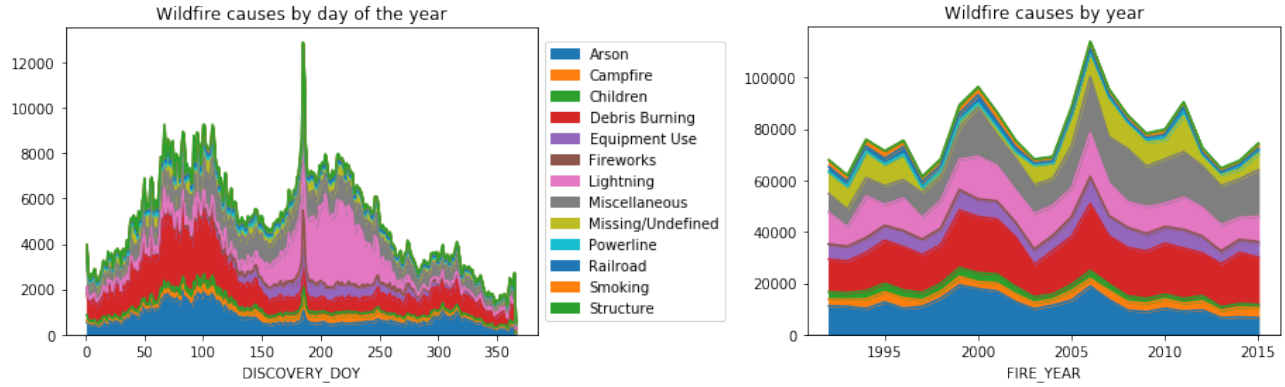


It is important also to get a sense of the distribution of the output values (fire causes):



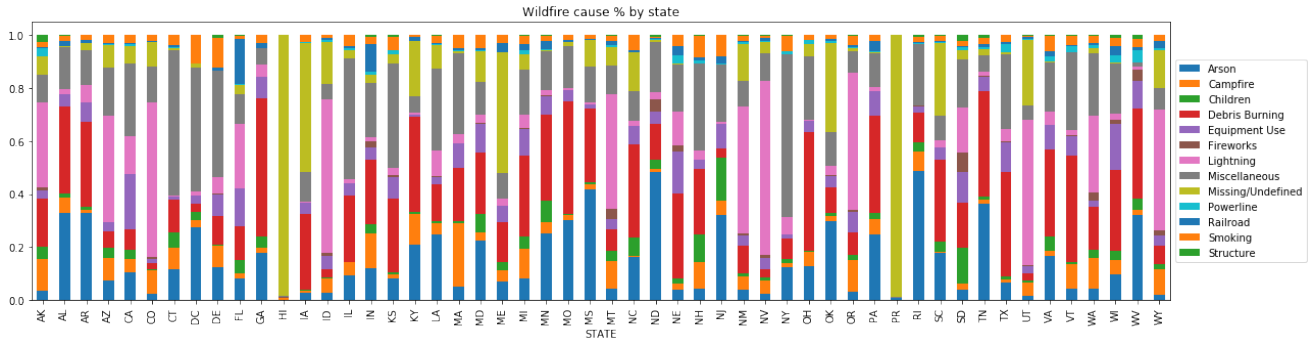
The most frequent cause is “Debris Burning”. The fact that these classes are very unbalanced means that the benchmark that always predicts the most frequent class is will have an accuracy which is greater than $1/N = 1/13 \sim 0.08$.

Further exploratory visualization can be done by looking at the evolution of these causes in time:

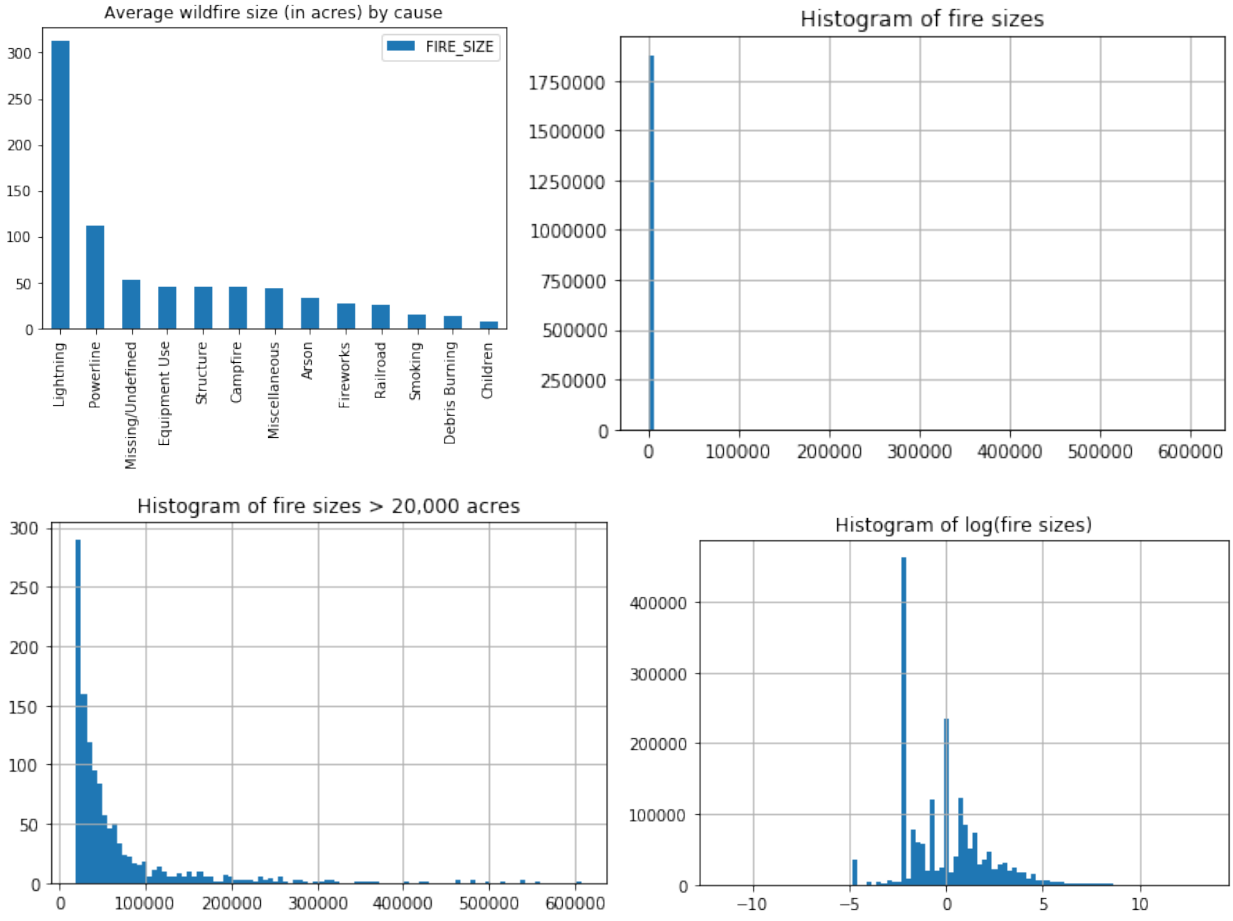


While the number of wildfires varies a lot year over year, their overall cause distribution does not seem to change that much. This distribution does appear to be closely tied to the time of year. Lightning fires for example, are more frequent in late Summer, and debris burning is more frequent in Spring.

On the other hand, cause distributions are very different by state:



Lightning fires are in average by far the largest. Most wildfires are relatively small, but some are very large, suggesting that a logarithmic scale is a reasonable transformation on the FIRE_SIZE feature.



Algorithms and Techniques

I need to produce models that estimate a wildfire's cause (from the above list of 13 possible causes) from its discovery date, size (in acres), and location (state). It is thus evident that I need to employ supervised learning algorithms.

From the analysis above we also see that these 13 classes are highly imbalanced in the data, with the majority class being 'Debris burning'. This is something to keep in mind when choosing the algorithms to train.

After some research I have decided to implement the following three algorithms: a linear model, XGBoost and an MLP (neural network). These models will be trained using Amazon SageMaker. More specific information follows.

1. Amazon SageMaker's **Linear Learner Algorithm** is a supervised learning algorithm that works on both regression and classification problems. In the classification use case, it learns a linear threshold function to predict the expected class.

2. **XGBoost** is a supervised gradient boosting algorithm that is meant to be relatively fast and accurate compared to other machine learning algorithms. It has proven to produce models with high accuracy in a number of fields. They may also be robust with unbalanced datasets.

3. **MLP** is another name for a neural network with dense layers. It is the most popular supervised learning algorithm and normally produces good results for large training sets.

Benchmark

Given that the input features are unlikely to uniquely define the output class in our classification problem, there is no expectation to obtain a very high accuracy. Instead my models should simply provide a reasonable guess for the cause of a wildfire given the input features.

I have decided to use a benchmark model which always picks the most frequent cause from the training data. That is 'Debris burning'. This benchmark produces an accuracy of 26% on my test set, and an F_1 score of 0.032.

III. METHODOLOGY

Data Preprocessing

1. NaN Values: The first step is to deal with missing (NaN) values or redundant records. Almost half of the rows in the data have at least one missing value, so simply omitting these rows is not an option. More specifically, the only column with missing values is **DISCOVERY_TIME**, with 878,703 NaN entries.

I came up with the simple solution to convert this column into a categorical feature with three classes: **Morning** (0500-1200), **Afternoon** (1201-1800), **Night** (1801-2400, 0000-0459), and **Unknown** (NaN). This new column was renamed as **DISCOVERY_PART_OF_THE_DAY**.

2. Target Column: The **STAT_CAUSE_CODE** values of 1 - 13 were mapped to 0 - 12, since Amazon SageMaker's classification models require the classes to start at 0.

So far, the pre-processed data-frame looks as follows:

	FIRE_YEAR	DISCOVERY_DOY	FIRE_SIZE	STATE	DISCOVERY_PART_OF_DAY	STAT_CAUSE_CODE
0	2005	33	0.10	CA	Afternoon	8
1	2004	133	0.25	CA	Morning	0
2	2004	152	0.10	CA	Night	4
3	2004	180	0.10	CA	Afternoon	0
4	2004	180	0.10	CA	Afternoon	0

3. Categorical Columns: The categorical columns **FIRE_YEAR**, **DISCOVERY_PART_OF_DAY** and **STATE** were one-hot encoded.

4. Fire Size: The **FIRE_SIZE** column has values that go up to more than half a million acres, yet most values are much smaller, as appreciated in the Exploratory Visualization section above. Thus the natural logarithm was applied to it, reducing its range between -15 and 15, and producing a more "normal" distribution.

5. DISCOVERY_DOY (Cyclical Value): This column could be considered numerical as is. However, doing so would mean that late December (~365) would be considered to be very far from early January (~1). Instead I map the values of this column to a unit circle, to account for its cyclic nature. The result is two columns being the **cosine** and **sine** of $2\pi \times$ **DISCOVERY_DOY** $/ 366$**.**

6. Dataset Split: The data is split into a training set, a validation set, and test set, including years 1995-2011, 2012-2013, 2014-2015, respectively. I chose to avoid a random split because real world test data will be to the future of the training set.

Implementation

All the models were trained on Amazon SageMaker. The Jupyter Notebooks included with this project contain the code necessary to train and deploy each model. Some of the implementation details are specified below for each model.

Linear Learner:

To define the multi-class linear estimator I use the default hyper-parameters, and specify:

```
balance_multiclass_weights=True,

epochs = 50
```

The 'balance_multiclass_weights' parameter means that classes are weighted to make up for the existing imbalance.

XGboost model:

For this model, Amazon SageMaker uses the open source XGBoost library.

For this algorithm I use hyper-parameter tuning and select the best model out of some models in the hyper-parameter space defined below.

```
hyperparameter_ranges = {

'max_depth': IntegerParameter(3, 12),
```

```
'eta': ContinuousParameter(0.05, 0.5),
'min_child_weight': IntegerParameter(2, 8),
'subsample': ContinuousParameter(0.5, 0.9),
'gamma': ContinuousParameter(0, 10) }
epochs: 30
```

Besides these, the following hyperparameters are set:

```
silent=0,
objective='multi:softmax',
early_stopping_rounds=10,
num_round=30
```

MLP model:

For this model, Amazon SageMaker uses the open source SciKitLearn library's MLPClassifier class. This model was defined using the default hyper-parameters with the following changes:

```
hidden_layer_sizes=(50,),
learning_rate_init=0.005,
max_iter=100
```

Due to the relatively small number of features, I decided to use a single hidden layer, having a size between the number of input features and the number of output features.

Refinement

Model to refine:

I decided to refine the MLP model as it is the one that produced the best average F_1 scores (see the Results section below).

I did some experimentation changing the learning rate and the hidden layer size, but the results did not improve by any significant measure.

Grouped labels:

Notice that the classes Powerline, Equipment Use, Structure, Railroad, Children, Smoking, and Campfire have relatively low counts and F_1 scores (see Results section below). For this reason, I decided to unify these labels into a single label named "**Other**".

For this I generated a new training, validation and test datasets with the following cause codes:

```
Debris Burning      1           429028
Lightning           0           278468
Arson               2           281455
Miscellaneous       3           323805
Missing/Undefined   4           166723
Other               5           241874
```

This new target is in turn less unbalanced that the original one.

Other refinement:

The variations of learning rate, number of epochs and hidden layer size below were attempted on the MLP model with grouped target classes. No significant improvement was observed.

Model	Learning rate	Number of epochs	Hidden layer size
1	0.005	100	50
2	0.05	100	50
3	0.005	100	25
4	0.05	100	25
5	0.009	100	25
6	0.0001	100	25
7	0.005	200	25

Of these, Model #1 produced the better metrics. See the Results section for more details.

Interpretability:

These models may be further improved by carrying out a deep dive into the importance they assign to each feature when making a prediction. The effort of quantifying those importance measures is known as model interpretability.

Using the open source ELI5 library's PermutationImportance estimator, I have derived the following importance values for the 20 most important features:

Weight	Feature
0.0619 ± 0.0009	TX
0.0532 ± 0.0013	DISCOVERY_DOY_COS
0.0433 ± 0.0011	NY
0.0397 ± 0.0009	FIRE_SIZE
0.0333 ± 0.0011	GA
0.0286 ± 0.0006	Night
0.0274 ± 0.0013	Afternoon
0.0225 ± 0.0011	DISCOVERY_DOY_SIN
0.0191 ± 0.0012	CA
0.0149 ± 0.0004	Morning
0.0143 ± 0.0005	KS
0.0108 ± 0.0001	SC
0.0105 ± 0.0006	AL
0.0094 ± 0.0002	AR
0.0089 ± 0.0005	KY
0.0084 ± 0.0003	OK
0.0072 ± 0.0003	TN
0.0068 ± 0.0004	MO
0.0063 ± 0.0004	FL

More work on this is done in the Free-Form Visualization section below.

IV. RESULTS

Model Evaluation and Validation

The following F_1 scores and accuracy were obtained for each of the models trained on all the original 13 target classes:

	Linear Learner	XGBoost	MLP	Benchmark
Accuracy	0.0866	0.3902	0.3894	0.26

Labels	F1 scores			
	Linear Learner	XGBoost	MLP	Benchmark
Lightning	0.0	0.58	0.58	0.0
Equipment Use	0.0	0.04	0.03	0.0
Smoking	0.0	0.0	0.0	0.0
Campfire	0.0	0.11	0.08	0.0
Debris Burning	0.07	0.51	0.51	0.07
Railroad	0.03	0.04	0.0	0.0
Arson	0.0	0.27	0.27	0.0
Children	0.0	0.00	0.12	0.0
Miscellaneous	0.0	0.42	0.39	0.0
Fireworks	0.12	0.38	0.27	0.0
Powerline	0.7	0.0	0.0	0.0
Structure	0.0	0.0	0.0	0.0
Missing/Undefined	0.14	0.15	0.12	0.0
Average	0.082	0.193	0.183	0.006

From the tables above I can see that even though the accuracy for the MLP and the XGBoost model are around 39%, their average F_1 scores are around 18%, which better reflects the imbalance in the results.

After the label grouping specified on the Refinement section above, some manual hyperparameter tuning was carried out on an MLP model. The best results were those of Model #1 from the Other Refinement subsection. This model has an accuracy of 0.3902 (only slightly higher than the first model with the 13 labels) and an average F_1 score of 0.35 (a substantial improvement), as shown below:

Labels	F1 scores
	MLP (6 classes)
Lightning	0.612258
Debris Burning	0.495596
Arson	0.298165
Miscellaneous	0.376625
Missing/Undefined	0.098685
Other	0.257304
Average	0.357

Justification

The MLP model with the labels grouped into 6 classes has an accuracy (0.3902) and average F_1 (0.357) score which is substantially better than those of the benchmark. While this is nowhere near a 100% accurate model, it is important to take into account that even an expert would probably make a low confidence guess of a wildfire's cause from the chosen input features.

The problem that this model solves is thus producing a somewhat **educated guess** for the cause of a wildfire given those input features.

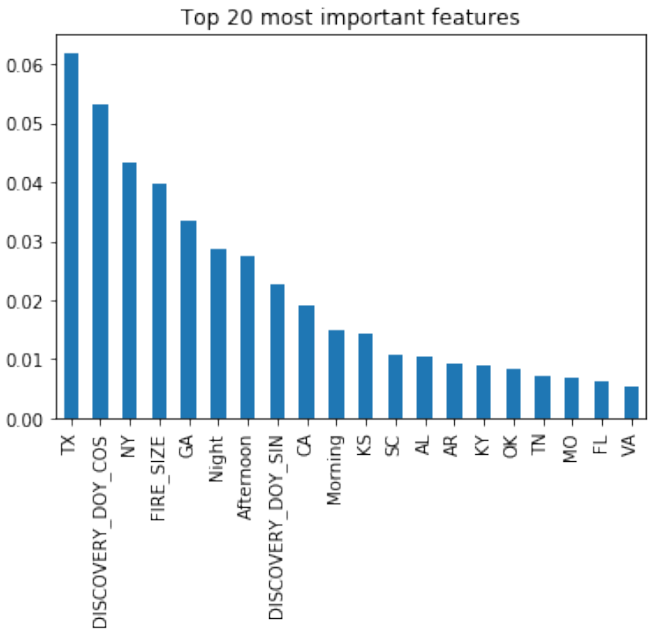
The **interpretability** measures above can also provide some insights into the relevance of each feature to make a wildfire cause prediction. More on this in the Free-Form Visualization section below.

V. CONCLUSION

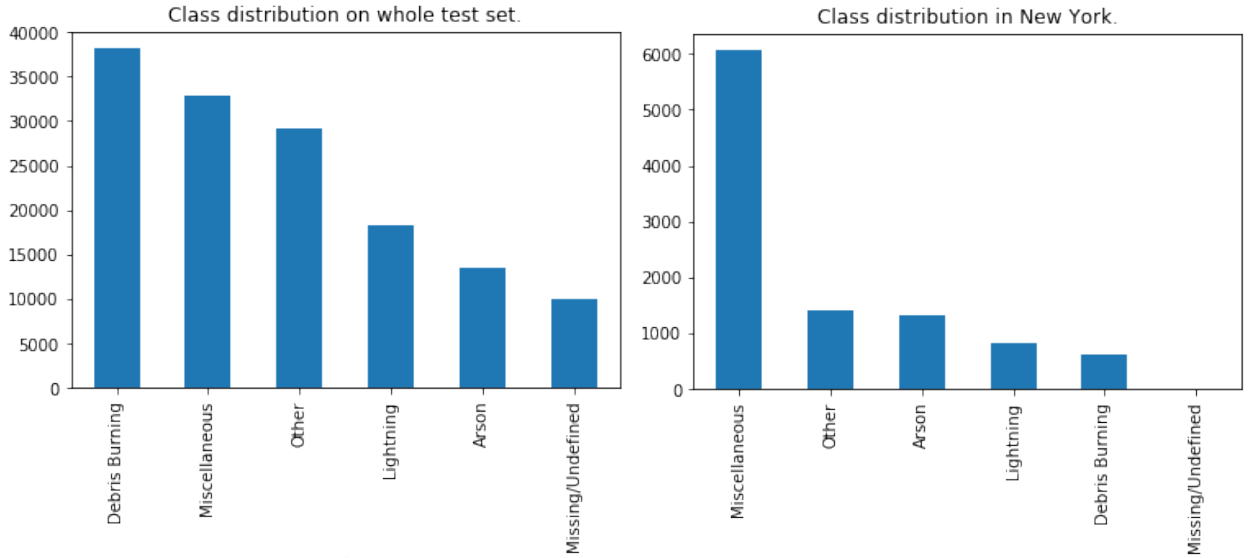
Free-Form Visualization (Model Interpretability)

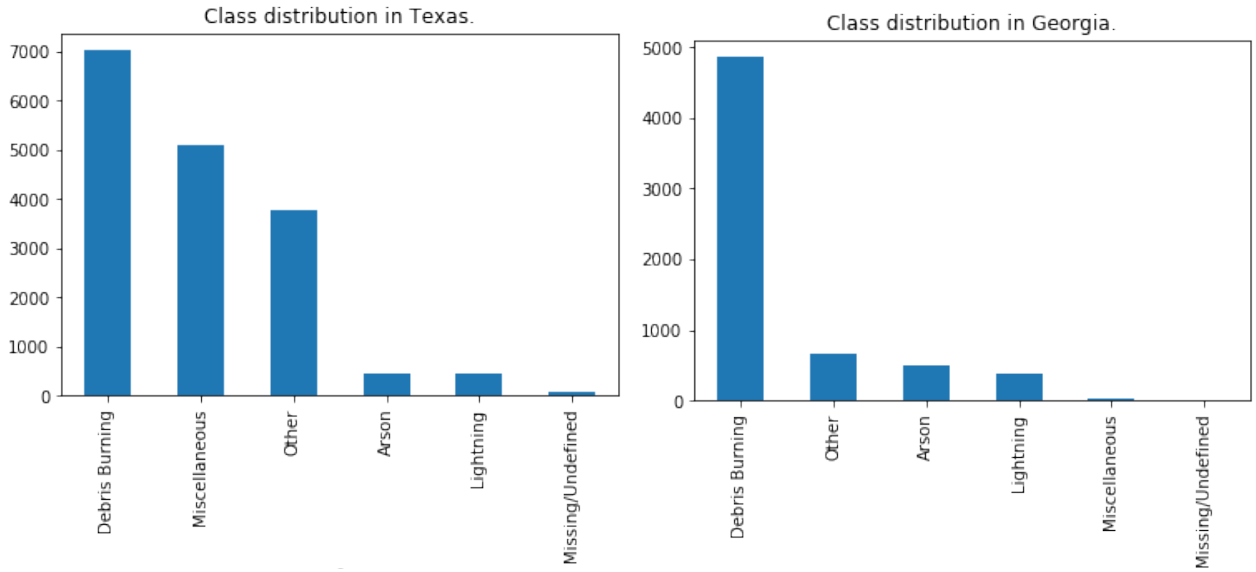
For this section I have chosen to dig a bit deeper into the Interpretability section results obtained above using the ELI5 library's PermutationImportance estimator.

The plot below charts feature importances of the 20 most important features:

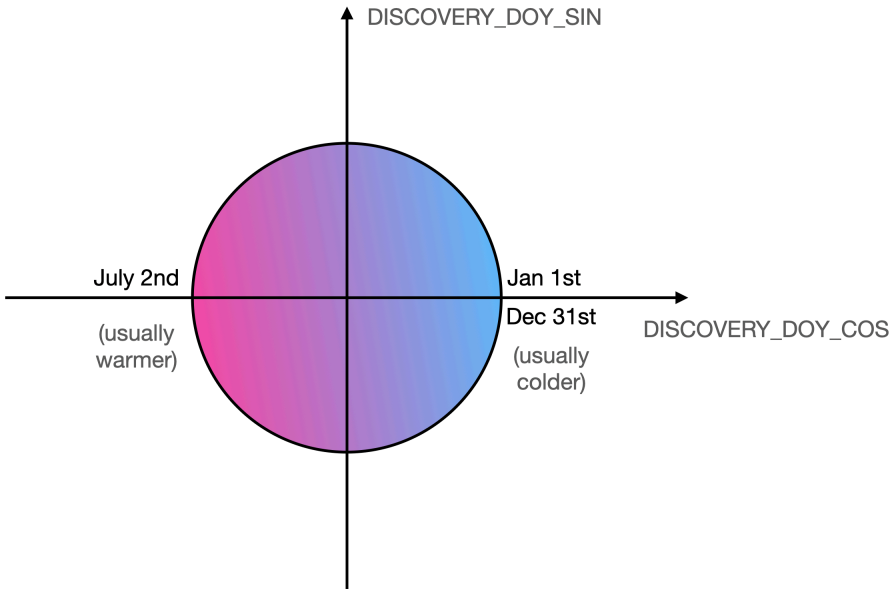


The appearance of states like TX, NY, and GA near the top suggests that these states have fire cause distributions which are substantially different from those of other states. Indeed, we can verify this fact by comparing these three states' cause distributions with that of the whole test set:





Another interesting observation is that DISCOVERY_DOY_COS is notably more important than DISCOVERY_DOY_SIN. My guess is simply that by placing DISCOVERY_DOY along the perimeter of a circle, DISCOVERY_DOY_COS represents the X axis, having positive values on days closer to New Year's, and negative values on days closer to July 2 (mid-year). In other words, DISCOVERY_DOY_COS is to some extent a measure of the average daily temperature on the day in question.



This suggests that the day's temperature may be a good candidate input feature for cause determination.

Reflection

I used the data from Kaggle's 1.88 Million US Wildfires dataset to train three different models, namely a linear model, an MLP model, and an XGBoost model to predict wildfire causes from their state, time of discovery, and size, under some feature transformations. I further improved the MLP model's accuracy by grouping some of the target labels, and I used interpretability tools to obtain and briefly analyze input feature weights. I did all this using Amazon Sage Maker to train, deploy, and test the models.

The accuracy and F_1 scores of the models trained in this project are not entirely satisfactory, in an area where we hear frequent brags of nearly 100% accuracy classification models into dozens of target classes. However, it is important to note that the chosen input features may simply not contain enough information for a full assessment of the cause of a wildfire.

I have nevertheless produced a model whose interpretable feature weights provide some insight into these input features which may be worth investigating further.

Amazon SageMaker turned out to be a great platform for training and deploying models, and this project has allowed me to obtain some hands on experience with its toolset.

Support Vector Machine did not make the cut into the project, even though it was mentioned in the proposal. The reason being that this estimator is expensive (slow) to train on such a large dataset.

Improvement

Some possible future improvements to this investigation are the following:

- Taking into consideration longitude and latitude rather than only the states would have likely yielded more accurate predictions, as these imply more specific information about the location of the wildfire.
- Fine tuning the MLP model under different geographic segments, as each state has a different cause distribution, would probably also improve predictions.
- Gathering more contextual data such as temperature or weather information may be fruitful, as suggested by the observation at the end of the Free-Form Visualization section.
- I believe more subject matter expertise is necessary in the path to getting better predictions. I would imagine that the features used here are not the only ones an expert would use to figure out the cause of a fire. Finding out what this training data could be useful to take this investigation to the next level.
- Narrowing down even more the wildfire causes to three classes such as *natural event, human activity and other* will increase the model scores.