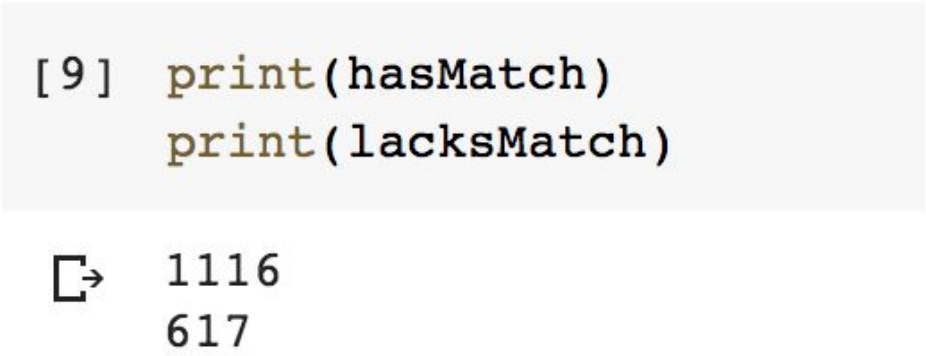
Code for Charlottesville Fire Risk Score Machine Learning Exploration Summary

One of the stated goals of this project was to use machine learning algorithms to predict a risk score for fires based on their location. This risk score would then be reported to firefighters when a call comes in from that location, to help them determine the appropriate units to send. We ultimately concluded that this idea is not executable given the current data for a number of reasons.

First, there's the question of standardizing addresses. Without a standardized address to unify data, we cannot merge historical data with information about the buildings. While our team made fantastic progress connecting data from across the Charlottesville Open Data Portal, and were able to develop techniques for merging contemporary data, these techniques cannot be extended to historical data. Not only do ways of writing addresses change over time, but the addresses themselves do, too. Out of 1,733 past fire incidents with information available, only 1,116, or 64.4%, were able to be matched to an address in the merged Open Data Portal dataset.



Code snippet reporting number of historical addresses with contemporary matches, and number of historical addresses which lack a contemporary match

Standardization and change over time present problems not only in merging data, but in the reliability of the data itself. Many past incidents lack any data entry for key outcomes, and property damage values are biased by inflation. While we can impute values for these fields to enable our models to run, this introduces an element of inaccuracy and guesswork into a model for which accuracy is crucial.

HEAT_SOURCE	ITEM_FIRST_IGNITED	CAUSE_OF_IGNITION	UNIT_1ST_ARRIVE	UNIT_1ST_RESP_TM_MIN	CIVILIAN_CASUALTY	FIRE SERVICE CASUALTY
NaN	NaN	Debris, vegetation burn	E7	4.85	NaN	NaN
NaN	NaN	Smoking	E4	3.35	NaN	NaN
Undetermined	Organic materials, other	Cause undetermined after investigation	E1	3.25	NaN	NaN
Undetermined	Undetermined	Cause under investigation	BC1	2.77	NaN	NaN
NaN	NaN	NaN	CHF2	4.67	NaN	NaN
Chemical reaction	Oily rags	Unintentional	BC1	4.43	NaN	NaN
Cigarette lighter	NaN	Misuse of fire	E7	3.37	NaN	NaN
Electrical arcing	Transformer, including transformer fluids	Failure of equipment or heat source	E7	4.35	NaN	NaN

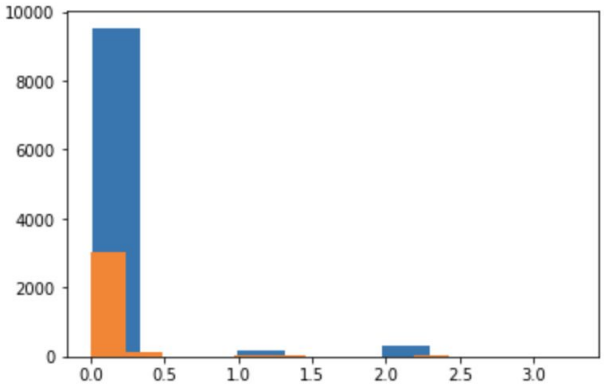
Random subset of the historical data with significant amount of NaN cells

Developing a label for historical data introduces another layer of complexity. The closer a label's distribution is to uniform, the more likely it is to be captured well by an algorithmic model. Unevenly distributed data is difficult to learn, and unfortunately the nature of fire outcomes is heavily unevenly distributed. Building fires worsen steadily until they reach a "flash point", after which the size and severity of the fire increases exponentially. Thanks to Charlottesville's dedicated fire department, few fires hit this point; however, the ones that do have orders of magnitude worse outcomes than the majority of historical fires. If we were to artificially smooth out the data in order to make it predictable with machine learning, we'd make that data less realistic, and our results significantly less useful.

```
import matplotlib.pyplot as plt

plt.hist(y_train)
plt.hist(y_test)
```

```
(array([3.013e+03, 1.490e+02, 1.100e+01, 3.000e+00, 5.600e+01, 3.800e+01,
        1.100e+01, 0.000e+00, 1.300e+01, 3.600e+01]),
 array([0.        , 0.243017  , 0.486034  , 0.729051  , 0.972068  ,
        1.215085  , 1.458102  , 1.70111899, 1.94413599, 2.18715299,
        2.43016999]),
 <a list of 10 Patch objects>)
```



Histogram showing the uneven distribution of risk scores, where risk is calculated as (normalized property damage + number of casualties)

Because the data is so unevenly distributed, simply guessing “low risk” gives a fairly low RMSE (root mean squared error) score. This means the RMSE of our models is misleadingly good. However, the RMSE of a model which literally does guess “low risk” regardless of the data actually outperforms our models. The chances that an incident is low risk are so high that it will never be profitable for a model to guess anything else; we can therefore conclude that machine learning cannot be applied to the current datasets available.

```
[32] from sklearn.model_selection import GridSearchCV

     gsc = GridSearchCV(
             estimator=SVR(kernel='rbf'),
             param_grid={
                 'C': [0.1, 1, 100, 1000],
                 'epsilon': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10],
                 'gamma': [0.0001, 0.001, 0.005, 0.1, 1, 3, 5]
             },
             cv=5, scoring='neg_mean_squared_error', verbose=0, n_jobs=-1)

     reg = gsc.fit(X_train, y_train).best_estimator_
     rmse = mean_squared_error(y_test, reg.predict(X_test))**.5

     rmse

     0.3840493644355618
```

Our best model achieves an RMSE of 0.384

```
[34] print(max_rmse)
     print(min_rmse)
     print(avg_rmse)

     2.2397623530675843
     0.384031975304451 77
     0.31935420144239 28
```

A model which always guesses the lowest possible risk score outperforms our model with an RMSE of only 0.319

Finally, we have to ask ourselves whether machines truly are better at identifying risk than experienced fire-fighters. Machine learning shines when there is a massive amount of reasonably clean and consistent data, too large for the patterns in the dataset to be identified by a human. However, machine learning is crippled by irregular data, as every model must be trained to a fixed set of inputs. Take, for example, hoarding flammable items. Before the global COVID-19 outbreak, this was a rare occurrence. A machine learning model trained on the past few years would not take hoarding into account and would fail to adapt to the changing behavior the pandemic inspired. On the other hand, experienced firefighters can identify this newly common risk factor as soon as it arises, and integrate it with their existing understanding of fire risk without needing to start over from scratch. Models are built for abstraction from large amounts of data; in this case, the specifics of what contributes to a risk score are as important as the score itself, and are lost in complex models.

Ultimately, a catalogue of identified risk factors would be extremely useful to maintain and to present when a call comes in, but a machine learning model that makes predictions based on that catalogue of risk factors would be much less so. There are already strides being made towards compiling a catalogue of this sort; if new entries can be matched to the address standardization used to merge the Open Data Portal datasets, then all that's needed to report these risk factors, as well as those in the Open Data Portal, is a UI (user interface). It is my recommendation based on an exploration of this data using machine learning that risk factors be reported separately, rather than compiled into a risk score, as a machine's abstraction from and conclusions based on this dataset will be of lower quality than the conclusions an experienced firefighter can come to based on the same information.