**Example Enron POI Identifier Report**

**Introduction**

As one of the largest corporate fraud cases in American history, the downfall of Enron is both fascinating and uniquely well-documented (thanks to the ensuing federal investigation). In this project, I use email and financial data for 146 executives at Enron to identify persons of interest in the fraud case. A person of interest (POI) is someone who was indicted for fraud, settled with the government, or testified in exchange for immunity. This report documents the machine learning techniques used in building a POI identifier.

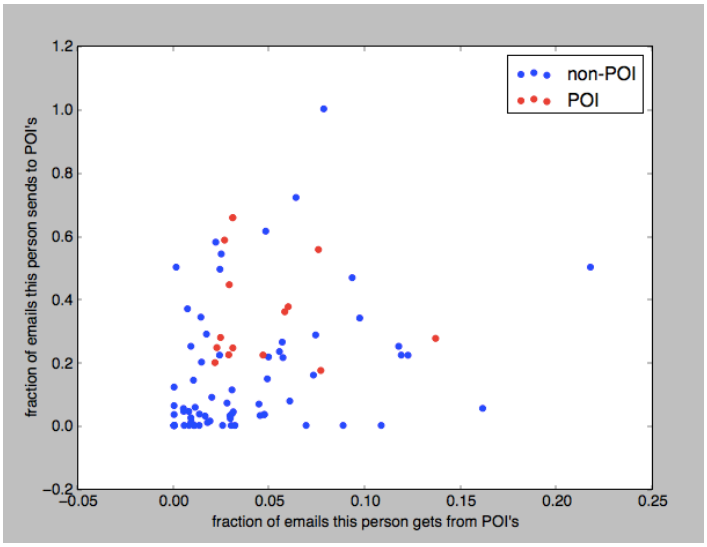
**The Enron Data**

The first step of this project is examining the data for any outliers or obvious mistakes. There was one significant outlier that arose from a spreadsheet summary line that got transcribed into the dataset; this outlier was removed by hand before proceeding. Two more financial outliers (Ken Lay and Jeff Skilling) were left in because they are clearly real Enron characters, and the fact that they made so much money off the company is not noise or a mistake--it's tightly entangled with the corporate fraud.

**Feature Processing**

Once the data was cleaned of outliers, the next step was selecting features to use. This was an iterative process, where a handful of features were selected for a first pass based on visualizations of the data that suggested that they might have some discriminatory power. These features were then used in the next steps of the process, selecting and evaluating a classification algorithm. Once a decision tree was selected as the algorithm for the POI identifier (further details on that process are given in the next section), the features were revisited for optimization. No feature scaling was deployed, as it's not necessary when using a decision tree.

Two new features were created and tested for this project. These were




- the fraction of all emails to a person that were sent from a person of interest
- the fraction of all emails that a person sent that were addressed to persons of interest

The hypothesis behind these features was that there might be stronger email connections between POIs than between POIs and non-POIs, and a scatterplot of these two features suggests that there might be some truth to that hypothesis.

The scatterplot of these features can be seen in the figure to the left.

In order to find the most effective features for classification, univariate feature selection was deployed to rank the features. The top 6 features were then selected for use in the final classifier; trying both smaller and larger numbers of features negatively impacted the precision and recall. These features were [**not going to give any hints here--not even saying that 6 is the right number, either!**]. As

can be seen from the list of features used in the classifier, the new fractional email features [were/were not] used in the final version because they [did/did not] help the precision [and/or] recall.

**Algorithm Selection and Tuning**
A decision tree classifier was selected as the algorithm for the POI identifier; Naive Bayes was also attempted (an SVM was found to be extremely slow to train, and abandoned).

The decision tree was selected over Naive Bayes because an examination of the Naive Bayes predictions seemed biased--almost all labels were 1.0 (POI), with only a few test events (out of several dozen) being labeled as 0.0 (non-POI).

After the algorithm and features were selected, the `min_samples_split` parameter of the decision tree was tuned by hand. The results of that tuning can be found in Table 1 below.

| `min_samples_split` | average precision | average recall |
|---|---|---|
| 2 | 0.44 | 0.5 |
| 3 | 0.40 | 0.33 |
| 4 | 0.38 | 0.44 |
| 5 | 0.43 | 0.33 |

*Table 1: Precision and recall when tuning the `min_samples_split` parameter.*

Since the best precision and recall were both found when `min_samples_split`=2, this was the clear choice for the value to set for this parameter.

**Analysis Validation and Performance**
This process was validated using 3-fold cross-validation. The precision and recall are somewhat different for each of the 3 folds, so these numbers were averaged and the average precision and recall were used as the final metric to quantify the performance of the algorithm.

As listed in Table 1, the average precision was 0.44 and the average recall was 0.5.

**Discussion and Conclusions**
The precision can be interpreted as the likelihood that a person who is identified as a POI is actually a true POI; the fact that this is 0.44 means that using this identifier to flag POI's would result in 56% of the positive flags being false alarms. Recall measures how likely it is that, given that there's a POI in the test set, this identifier would flag him or her; 50% of the time it would catch that person, and 50% of the time it wouldn't.

While these numbers are reasonably good, certainly better than guessing or a very unoptimized classification process, it would still be challenging to rely exclusively on this POI identifier alone to flag POIs. If either precision or recall were close to 1.0 then it would be easier to interpret as either (1) a very conservative algorithm that sometimes misses POI's but is rarely wrong when it does find one, or (2) a trigger-happy algorithm that rarely misses POI's but is prone to false alarms. An example of the latter is what we saw with the Naive Bayes classifier; the recall for that algorithm was 1.0 but only a handful of people were classified as non-POIs. While this could be useful, it seemed more interesting and valuable to attempt a classifier that showed less bias.

One thing that made this work challenging was that there were only 18 examples of POIs in the dataset. There were 35 people who were POIs in "real life", but for various reasons, half of those are not present in this dataset--by the "more data is better than a finely-tuned algorithm" maxim of machine learning, having a larger dataset is likely the single biggest way to improve the performance of this analysis.

Another possible path to improvement is digging in to the emails data more. The email features in the starter dataset were aggregated over all the messages for a given person. By digging into the text of each individual's messages, it's possible that more detailed patterns (say, messages to/from a specific address, rather than just messages to/from any POI address, or the usage of specific vocabulary terms) might emerge. Since we live in a world in which more POI finance data might not be easy to find, the next realistic thing to try might be to extract more data from the emails.