by Daria Alekseeva

Project Overview

In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, there was a significant amount of typically confidential information entered into public record, including tens of thousands of emails and detailed financial data for top executives.

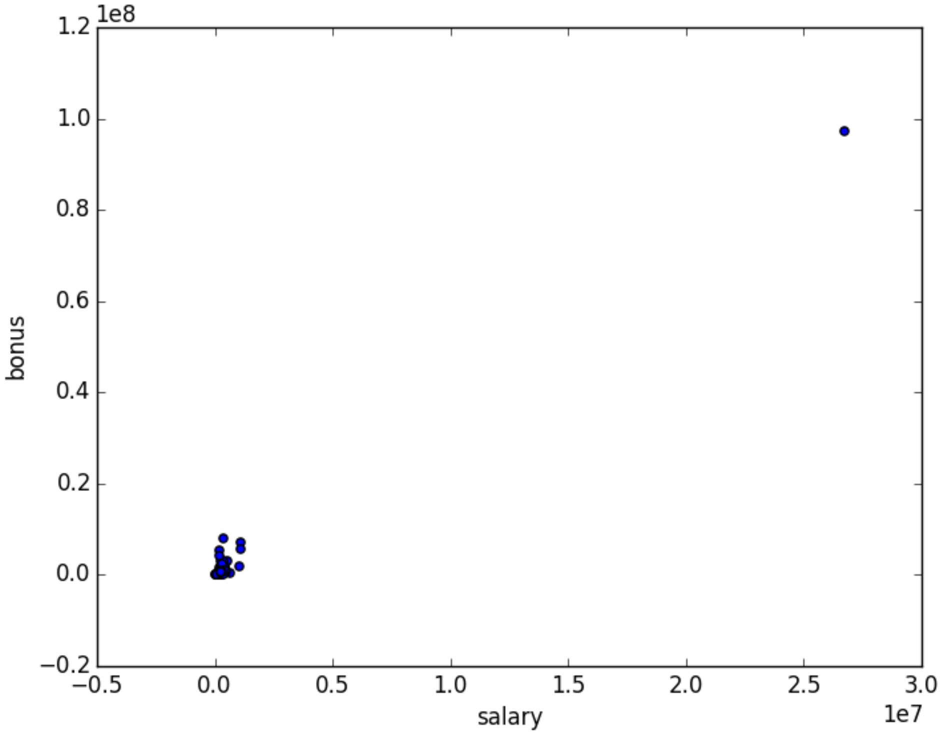
In this project I will build a person of interest identifier based on financial and email data made public as a result of the Enron scandal. I use email and financial data for 146 executives at Enron to identify persons of interest in the fraud case. A person of interest (POI) is someone who was indicted 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.

There are four major steps in my project:

1. Enron dataset
2. Feature processing
3. Algorithm
4. Validation

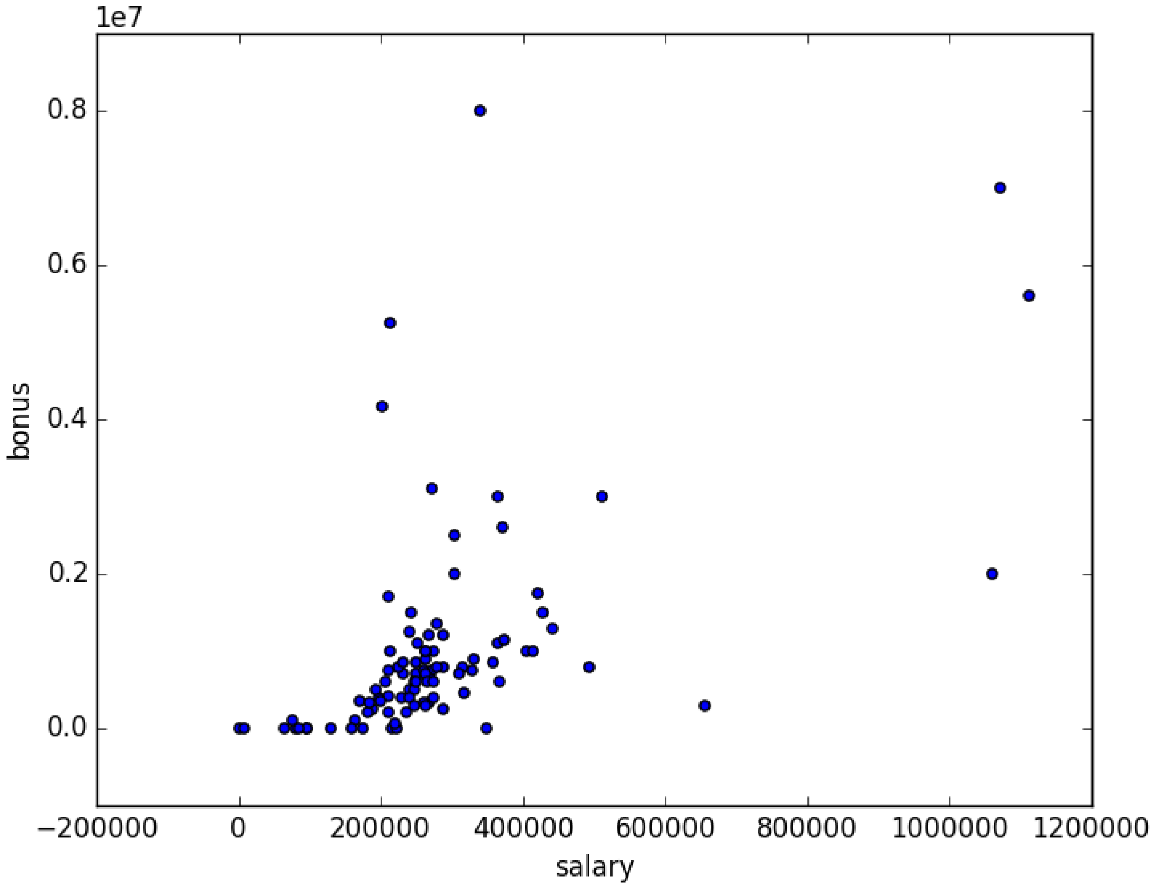
The Enron Data

First of all I'd like to have a look at my data and check it for outliers. I plot salaries and bonuses on Enron employees and see an outlier in the data.



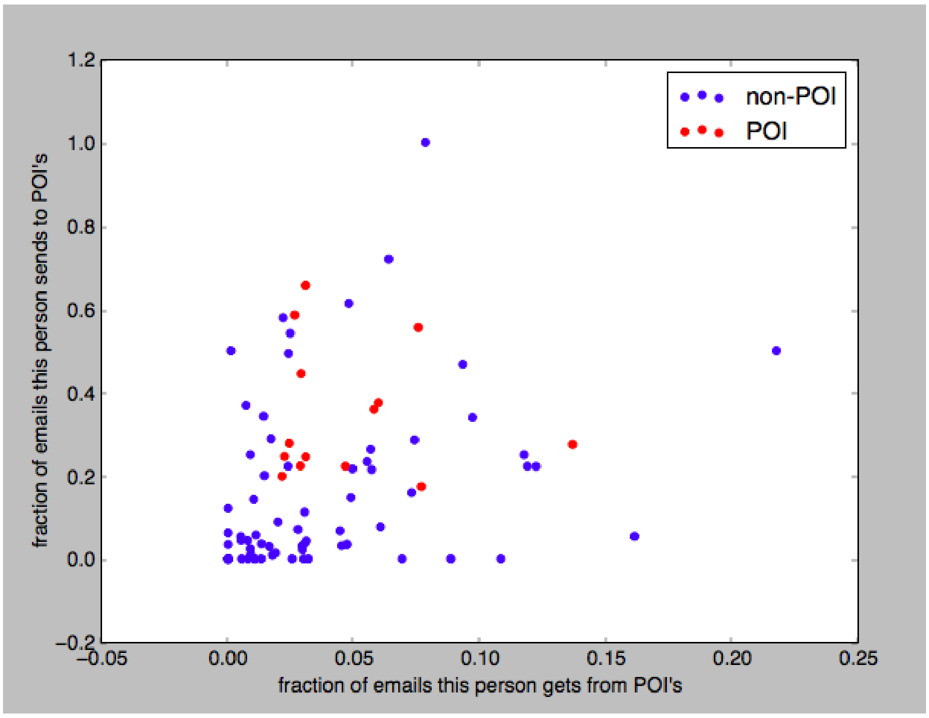
When I check it I see this is a number for total salary and bonus. As this is not sensible information for our analysis I remove it manually. Two more outliers (SKILLING JEFFREY and

LAY KENNETH) I keep in dataset as these values real and actually they are already a sign of these two managers being involved in the fraud. Now dataset look like this:



Feature processing

After cleaning the data from outliers I had to pick the most sensible features to use. First I picked 'from_poi_to_this_person' and 'from_this_person_to_poi' but there is was no strong pattern when I plotted the data so I used fractions for both features of “from/to poi messages” and “total from/to messages”.



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.

My hypothesis was that there is stronger connection between POI's via email that between POI's and non-POI's. When we look at scatterplot we can agree that the data pattern confirms said above, e.i. there is no POI below 0.2 in “y” axis.

In order to find the most effective features for classification, feature selection using “Decision Tree” was deployed to rank the features. Selection features was half manual iterative process. First I put all the possible features into features_list and then started deleting them one by one using score value and human intuition.

I picked 10 features which are:
[**"salary"**, **"bonus"**, **"fraction_from_poi_email"**, **"fraction_to_poi_email"**, **'deferral_payments'**, **'total_payments'**, **'loan_advances'**, **'restricted_stock_deferred'**, **'deferred_income'**, **'total_stock_value'**]

Accuracy for this feature set is around 0.8.

Approximate **feature ranking**:
1 feature salary (0.211707133446)
2 feature bonus (0.14622972935)
3 feature fraction_from_poi_email (0.120901730257)
4 feature fraction_to_poi_email (0.118337314859)
5 feature deferral_payments (0.0955181169023)
6 feature total_payments (0.0879795396419)
7 feature loan_advances (0.0747826086957)
8 feature restricted_stock_deferred (0.0534161490683)
9 feature deferred_income (0.0534161490683)
10 feature total_stock_value (0.0377115287109)

But with these features my precision and recall were too low (less than 0.3) so I had to change my strategy and manually pick features which gave me precision and recall values over 0.3. In this dataset I cannot use accuracy for evaluating my algorithm because there a few POI's in dataset and the best evaluator are precision and recall. 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.

Finally I picked the following features:
["fraction_from_poi_email", "fraction_to_poi_email", “shared_receipt_with_poi”]

Algorithm Selection and Tuning

Firstly I tried Naive Bayes accuracy was lower than with Decision Tree Algorithm (0.83 and 0.9 respectively). I made a conclusion that that the feature set I used does not suit the distributional and interactive assumptions of Naive Bayes well.

I selected Decision Tree Algorithm for the POI identifier. It gave me accuracy before tuning parameters = 0.9. No feature scaling was deployed, as it's not necessary when using a decision tree.

After selecting features and algorithm I manually tuned parameter **min_samples_split**.

min_samples_split	*precision*	*recall*
2	0.67	0.8
3	0.57	0.8
4	0.57	0.8
5	0.8	0.8
6	0.8	0.8
7	0.67	0.8
average	0.68	0.8

It turned out that the best values for min_samples_split are 5 and 6.

Analysis Validation and Performance

This process was validated using 3-fold cross-validation, precision and recall scores.

First I used accuracy to evaluate my algorithm. It was a mistake because in this case we have a class imbalance problem - the number of POIs is small compared to the total number of examples in the dataset. So I had to use precision and recall for these activities instead.

I was able to reach average value of precision = 0.68, recall = 0.8.

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 0.68 means that using this identifier to flag POI's would result in 32% of the positive flags being false alarms. Recall measures how likely it is that identifier will flag a POI in the test set. 80% of the time it would catch that person, and 20% of the time it wouldn't.

These numbers are quite good but we still can improve the strategy. One of the possible paths 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.