Identify Fraud from Enron emails

Author : Vijayasarathi Balasubramanian

Project Overview

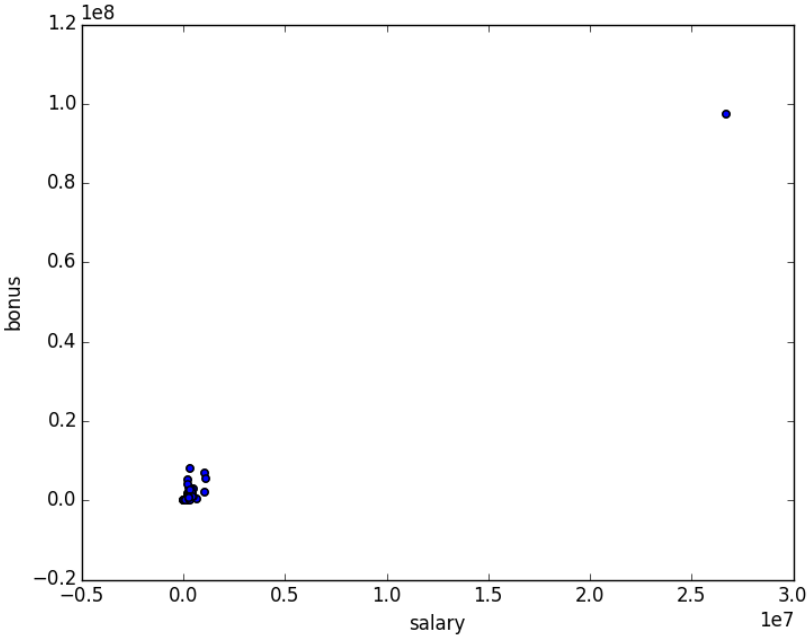
In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, there was a significant amount of typically confidential information entered into public record, including tens of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

In this project, we are going to play detective role, and use our skills to build a person of interest identifier based on financial and email data made public as a result of the Enron scandal. To start the detective work, we've took the data that's combined with a hand-generated list of persons of interest in the fraud case, which means individuals who were indicted, reached a settlement, or plea deal with the government, or testified in exchange for prosecution immunity.
There are four major steps in my project:

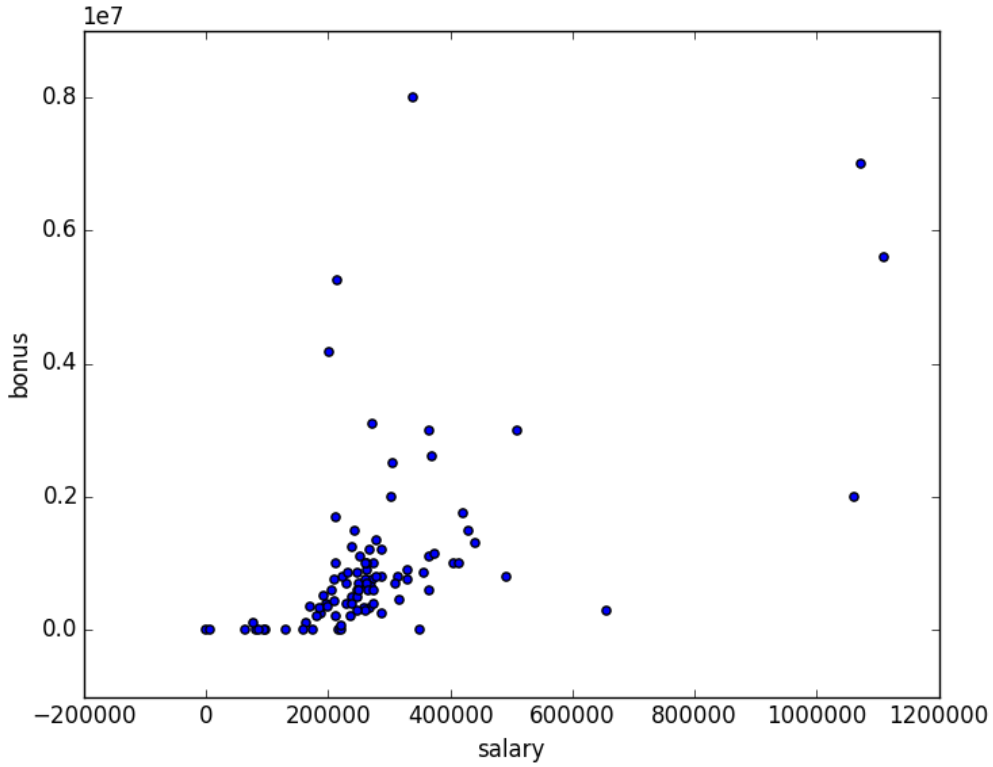
1. Enron dataset understanding.
2. Feature processing
3. Machine learning Algorithm selection.
4. Discussion and conclusion.

1. Enron Data set understanding:

Before we impose Machine learning algorithams on the data set we need to understand the data set bit to see if we have any outliers or issues in the data set, so that we can rely on the results.
To start with lets plot the salaries vs bonus.



from the above plot it looks like a number for total salary and bonus. As this is not giving us much information for our analysis lets remove the outliers. Two more outliers (SKILLING JEFFREY and LAY KENNETH) I am leaving in the 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:

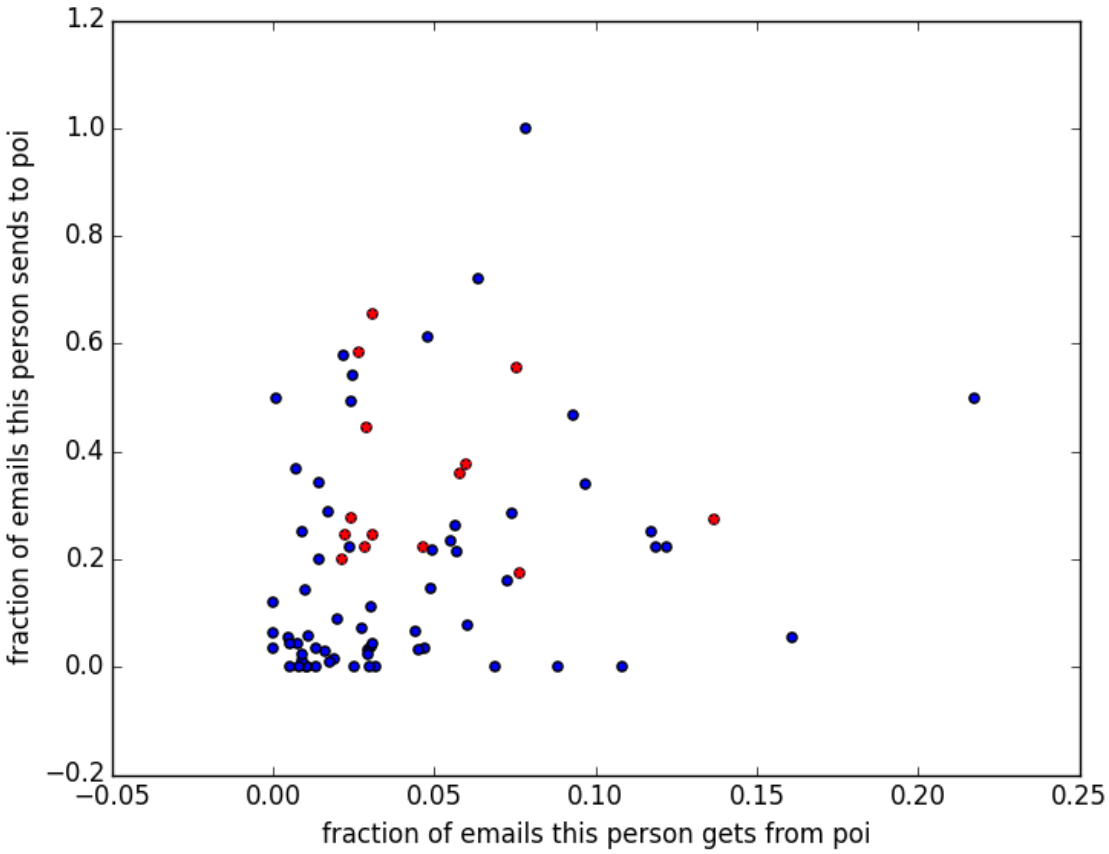


2. Feature Processing:

After removing the outliers i tried to use the features that gives meaningful insights. when we tried to use 'from_poi_to_this_person' or 'from_this_person_to_poi' it didnt give any interesting pattern. then i tried with of “from/to poi messages” and “total from/to messages”.

Identify Fraud from Enron emails

Author : Vijayasarathi Balasubramanian



Two new features were created and tested for this project.

1) The fraction of all the emails to a person that were sent from POI(person of interest).
2) Fraction of all emails that a person sent, that were addressed to POI(person of interest).

My Hypothesis is that there will be stronger connection between POI's via email that that of between POI's and non-POI's. when we look at the above scatter plot we can observe that the data pattern comes closure to our hypothesis. i.e:there is no poi's below .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. Feature selection process involves little bit of manual iterations.first I put all the possible feature_list and then stated deleting them one by one using score value and my own intuition.

I have selected 10 features :

"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 0.75.

Approximate feature ranking:

1 feature salary (0.211707133446)
2 feature bonus (0.195434931454)
3 feature fraction_from_poi_email (0.14622972935)
4 feature fraction_to_poi_email (0.118337314859)
5 feature deferral_payments (0.0879795396419)
6 feature total_payments (0.0747826086957)
7 feature loan_advances (0.0534161490683)
8 feature restricted_stock_deferred (0.0534161490683)
9 feature deferred_income (0.0377115287109)
10 feature total_stock_value (0.0209849157054)

But with these features my precision and recall were too low (less than 0.25) so I had to change my strategy and manually pick features which gave me precision and recall values over 0.25. 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"]

3. Machine Learning Algoritham Selection :

First 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 enough. 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 we use a decision tree. After selecting features and algorithm I manually tuned parameter min_samples_split.

Identify Fraud from Enron emails

Author : Vijayasarathi Balasubramanian

min_samples_split	precision	recall
2	0.5	0.67
3	0.67	0.67
4	0.67	0.67
5	0.67	0.67
6	0.5	0.67
7	0.5	0.67
average	0.585	0.67

It turned out that the best values for min_samples_split are 4 and 5.

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 = 0.58, recall = 0.67.

4. Discussion and conclusion:

The precision can be interpreted as the likelihood that a person who is identified as a POI is actually a true POI(Person of Interest). The fact that this is .585 which means using precision to flag a POI's would result in 41.5% of the positive flags being false alarms.

Recall measures how likely it is that identifier will flag as POI in the test. 66.667% of the time it would catch that person and 20% of the time it would not.

These numbers are coming closer but we can definitely improve these. One possibility is that we read more emails though the algorithms and understand the emails better. May be we need to read the raw email text or go in detail and understand the words in the emails. Based on the words and the context in which words used we can improve and predict better.

Also, instead of reading emails POI's vs non POI's, we can just read the emails and predict the trend in the email set across all the emails.