Identify Fraud from Enron Email

Nicole Mister

Enron, an energy commodities and services company, declared bankruptcy in 2001. The company then went under investigation for fraud. A corpus of emails was used in the investigation to determine persons of interest (POI) in the case. The goal of this project is to use this corpus of emails as well as employee financial data to create a machine learning algorithm to predict if an Enron employee is a person of interest.

Our dataset contains 146 records with 21 features. Of the 146 records, 18 are POI. After graphing a scatterplot of salaries and bonuses, it was clear that there were some outliers. After a quick visual inspection of enron61702insiderpay.pdf, I was able to determine "TOTAL" and "THE TRAVEL AGENCY IN THE PARK" were outliers that should be removed. I used the dicitionary pop function to remove those records. While another scatterplot indicated that the data still included additional outliers, I did not remove any additional data. The outliers included several POIs. We would therefore potentially lose pertinent data by removing these points. Additionally, by reviewing the PDF, I was able to determine several features that did not contain much data: loan_advances, director_fees, restricted_stock_deferred, deferral_payment. Additionally, the dataset contained email addresses, which in addition to not converting nicely into strings, is unlikely to contain data relevant to locating a POI. I removed these features and used 16 of the provided features for first set of algorithms.

Because I believe the proportion of emails to and from a POI may be more relevant than just the sum of emails, I created two new features, from_ratio and to_ratio. The to_ratio is the number of emails from this person to a POI divided by the total number of emails sent by this person. The from_ratio is the number of emails from this person to a POI divided by the total number of emails from this person.

To select features, I used SelectKBest to select the best 10 features. Keeping the features to only the most relevant to POI identification. The scoring is as follows:

```
salary 18.2896840434
total_payments 8.77277773009
bonus 20.7922520472
deferred_income 11.4584765793
total_stock_value 24.1828986786
exercised_stock_options 6.09417331064
long_term_incentive 24.8150797332
restricted_stock 4.187477507
shared_receipt_with_poi 9.92218601319
from_ratio 9.21281062198
```

I did not use scaling for these features.

Two algorithms met the 0.30 or better precision and recall score requirement of this project: Decision Tree and Naïve Bayes. The Decision Tree algorithm score is slightly better with 0.345 precision and 0.333 recall. I also tried KNeighbors. KNeighbors had higher precision (0.639), but lower recall (0.168).

Tuning the parameters means to adjust settings of the algorithm which may impact how well the algorithm performs on a dataset. The Decision Tree algorithm's precision and recall dropped (0.153 and 0.037 respectively) when the minimum samples split was set to 100. When it was

Identify Fraud from Enron Email

Nicole Mister

set to 10, the precision increased (0.370) but recall decreased (0.299). Setting the criterion to entropy slightly improved the precision (0.365) and lowered the recall (0.297). Setting the max features to auto slightly improved both precision (0.345) and (0.333). Ultimately, I only used max features.

Validation tests an algorithm's performance against a set of data not used in the training. If validation is not performed correctly, for instance, testing on the same data used to train, you can overtrain your data so that the effectiveness of the algorithm seems better than it actually is. I validated my analysis through cross validation and a test size of 0.3.

The evaluation metrics I used to evaluate the algorithms in this analysis were precision and recall. In this case, I think that it would be important to have a high recall since a false positive can be ruled out with other evidence, but missing a POI might mean the individual is never identified.