Data Exploration

Goals

My primary objective is to create a model that can identify all possible persons of interest while decreasing the number of false positives as much as possible. Machine learning can go through the data and pick up patterns based on the people that are known persons and interest and look for similar patterns in other people. Machine learning is useful here because of the amount of data is big enough so that it would be both impractical and inaccurate if done manually.

Data Summary

The data contains emails and financial data for 146 people. The main motivation behind the fraud at Enron was money, so financial data can be used to identify those that would benefit most from the fraud. Email data can be used to analyse communications of persons and interests and to see if any other people exhibit similar behaviour.

- 146 data points in data set
    - Persons of interest: 18
    - Non persons of interest: 128
- 21 features
- Features are divided into 3 main categories:
    - Financial (all units are in US dollars)
        - salary
        - deferral_payments
        - total_payments
        - loan_advances
        - bonus
        - restricted_stock_deferred
        - deferred_income
        - total_stock_value
        - expenses
        - exercised_stock_options
        - other
        - long_term_incentive
        - restricted_stock
        - director_fees
    - Email (all number of emails, except for email_address which is a text string)
        - to_messages
        - email_address
        - from_poi_to_this_person
        - from_messages
        - from_this_person_to_poi
        - shared_receipt_with_poi
    - POI (Boolean value indicating person of interest)
        - POI

Missing Data

Table 1 shows the number of records that are missing data for a specific feature. Data is assumed missing if the value is “NaN”. Features with a high count of missing data are not likely to be suitable to be used as features.

Note that there are 6 features which are missing data for over half of the dataset. These are all financial features that occur very rarely and is therefore only relevant for a selected number of indviduals.

FEATURE	MISSING COUNT
LOAN_ADVANCES	142
DIRECTOR_FEES	129
RESTRICTED_STOCK_DEFERRED	128
DEFERRAL_PAYMENTS	107
DEFERRED_INCOME	97
LONG_TERM_INCENTIVE	80
BONUS	64
TO_MESSAGES	59
SHARED_RECEIPT_WITH_POI	59
FROM_POI_TO_THIS_PERSON	59
FROM_MESSAGES	59
FROM_THIS_PERSON_TO_POI	59
OTHER	53
SALARY	51
EXPENSES	51
EXERCISED_STOCK_OPTIONS	44
RESTRICTED_STOCK	36
EMAIL_ADDRESS	34
TOTAL_PAYMENTS	21
TOTAL_STOCK_VALUE	20

Table 1 Number of times features had missing data

Outliers

There was a person “Total” key in the dictionary which had extremely large values. This value is the total row in the financial data, it was removed from the python dictionary before any other processing.

Features

New Features

Two new features were created:

- from_poi_to_this_person_ratio = Fraction of all emails to a person that were sent from a person of interest
- from_this_person_to_poi_ratio =Fraction of all emails that a person sent that were addressed to persons of interest

These features would give me a better understanding on the relative amount of emails that were exchanged with persons of interest. I also think if a person of interest would be interacting with another person of interest more than others.

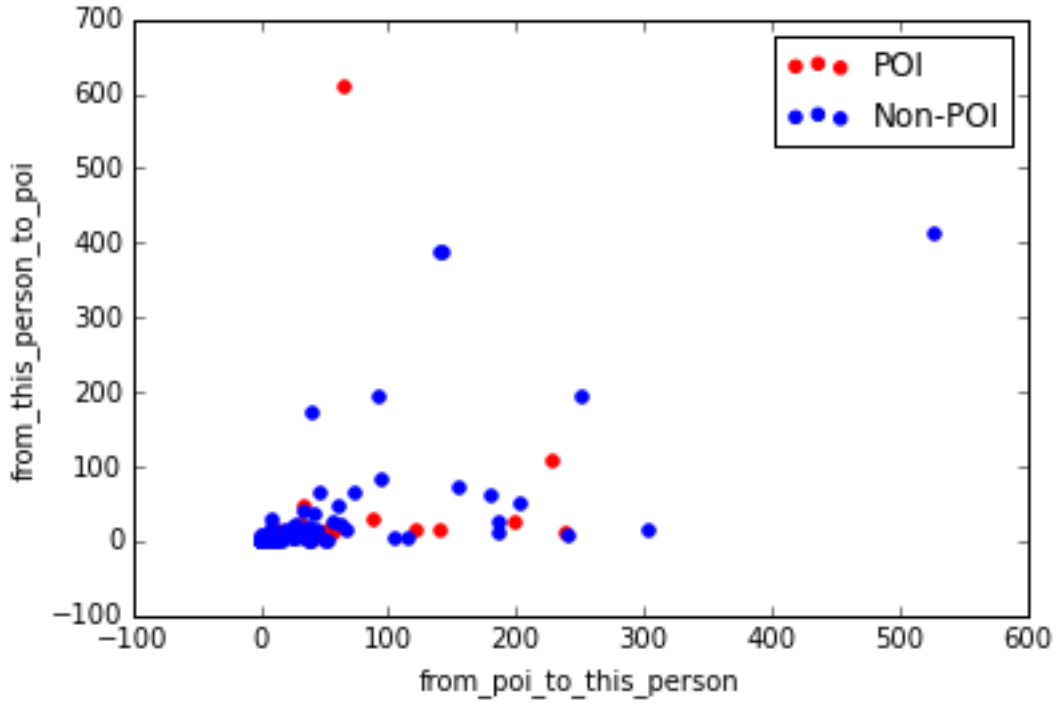

Chart 1 Emails from or to a person of interest classified by known persons of interest

In Chart 1 the data between 0 - 50 from_poi_to_this_person is overplotted. To improve the plot I implemented the 2 features discussed above, and I end up with this:

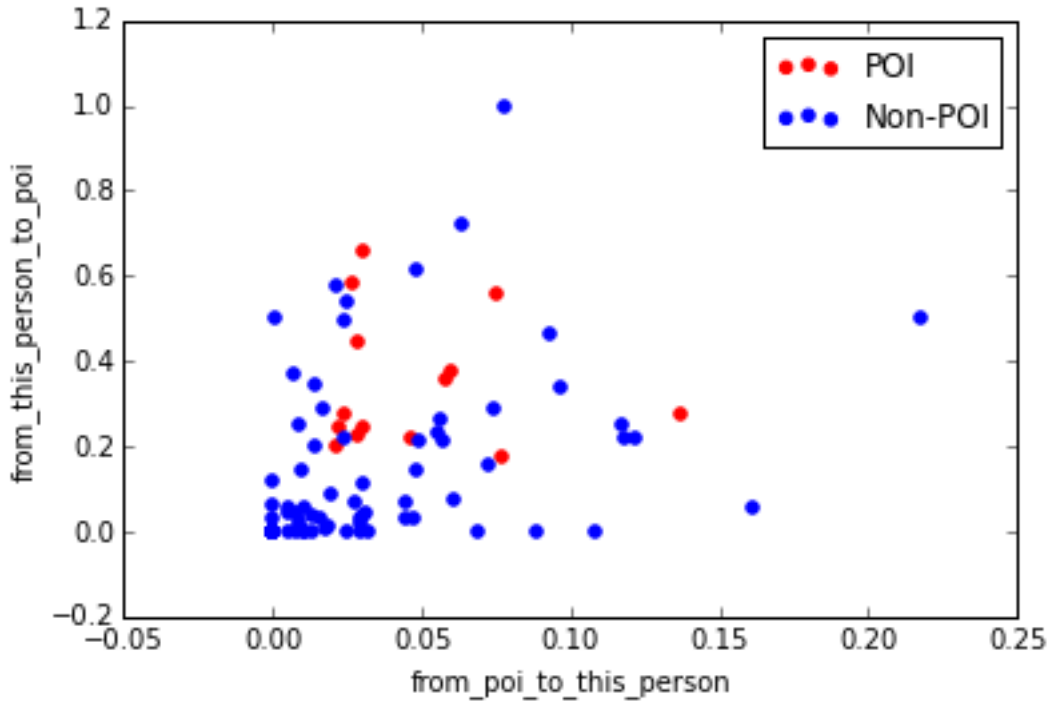

Chart 2 Ratio of emails from or to a persons of interest classifed by known persons of interest

Chart 2 shows some evidence that persons of interests may have more email interaction with one another.

Effect on Algorithm Performance

To test the effect of these new features on performance, I ran a decision tree classifier (more details on algorithm selection to follw) using all features as inputs. I then removed all features with lower importance then the from_poi_to_this_person or from_this_person_to_poi features. Here are the metrics of the decision tree model before including these 2 new features.

Accuracy	Precision	Recall	F1	F2
0. 80047	0. 80047	0. 22100	0. 22801	0. 22375

Table 2 Decision tree metics without new features

Importance:

- from_poi_to_this_person (0.035178)
- from_this_person_to_poi (0.057826)

After including from_poi_to_this_person_ratio. There is a drop across all metrics, this new feature does not improve the model.

Accuracy	Precision	Recall	F1	F2
0. 79820	0. 22437	0. 20900	0. 21641	0. 21190

Table 3 Decision tree metrics after adding from_poi_to_this_person_ratio

Importance: from_poi_to_this_person_ratio (0.021825)

And, after including from_this_person_to_poi_ratio. There is a large improvement in both precession, recall and importance of the new feature. This feature improves the model.

Accuracy	Precision	Recall	F1	F2
0. 82447	0. 33558	0. 32300	0. 32917	0. 32544

Table 4 Decision tree metrics after add from_this_person_to_poi_ratio

Importance: from_this_person_to_poi_ratio (0.150965)

Feature Selection Process Goals

My primary objective is to create a model that can identify all possible persons of interest while decreasing the number of false positives as much as possible. Machine learning can go through the data and pick up patterns based on the people that are known persons and interest and look for similar patterns in other people. Machine learning is useful here because of the amount of data is big enough so that it would be both impractical and inaccurate if done manually.

Recursive feature selection was implemented where I modified tester.py so that it would output the average feature importance of each iteration of the test of the model. I started to remove features one by one from those with the least importance and looked at how it affected the metrics for the model. I stopped removing features when I saw some large decreases in the metrics when a feature was removed.

There was very little effect on the metrics on my earlier iterations

All Features Included

Accuracy	Precision	Recall	F1	F2
0.81640	0.30863	0.30400	0.30630	0.30491

Table 5 Decision Tree metrics with all features

Removed Weak Features

- to_messages
- from_messages
- directors_fees
- loan_advances
- deferral_payments
- restricted_stock_deferred

Accuracy	Precision	Recall	F1	F2
0.82133	0.32724	0.32200	0.32460	0.32303

Table 6 Decision tree metrics with weak features removed

Final Features

I kept on iterating using this method until I was left with 4 features (average importance included in brackets):

- exercised_stock_options (0.289734)
- expenses (0.267030)
- shared_receipt_with_poi (0.231217)
- other (0.212019)

Accuracy	Precision	Recall	F1	F2
0.83429	0.42537	0.45600	0.44015	0.44953

Table 7 Decision tree metrics with final features

If the feature with the lowest importance is removed (other):

Accuracy	Precision	Recall	F1	F2
0.81043	0.36191	0.42850	0.39240	0.44953

Table 8 Decision tree metrics with "Other" feature removed

Note the large drop in precision, and moderate drops in accuracy and recall. Therefore, I decided to keep all of these features.

Feature Scaling

No feature scaling was deployed as it is not necessary for decision trees.

Algorithms Selection and Tuning

Naïve Bayes, SVM and decision tree classifiers were attempted. SVM was extremely slow to train and was abandoned without result.

A few iterations of Naïve Bayes and decisions trees was attempted. I found that Naive Bayes had inferior recall and precision to the decision tree classifier. Precision was similar across both, this was expected because of the large number of non persons of interest in the data, so it is not really a good indication of the quality of the model.

CLASSIFIER	ACCURACY	PRECISION	RECALL	F1	F2
NAÏVE BAYES	0.83529	0.35059	0.17950	0.23743	0.19891

DECISION TREE	0.83479	0.42711	0.45850	0.44225	0.45186

CLASSIFIER	TOTAL PREDICTIONS	TRUE POSITIVES	FALSE POSITIVES	FALSE NEGATIVES	TRUE NEGATIVES
NAÏVE BAYES	14000	359	665	1641	11335
DECISION TREE	14000	917	1230	1083	10770

Naïve Bayes is not as suitable as decision trees for this dataset because the sample size is small and the number of persons of interest is only 18. Since Naïve Bayes works off probabilities the sample of persons of interest is too small to accurately identify persons of interest (depending on the test/training data split this number will be even smaller).

Naïve Bayes had a very low recall, it was very good at identifying negatives (non persons of interest) but it was very bad at identifying positives.

Given, the performance of these algorithms decision tree was selected.

Parameter Tuning

Some algorithms take in parameters which change how it behaves, thus, changing the results. Parameters are used so that algorithms can adapt to different datasets and situations.

Since a decision tree was selected, the criterion and min_samples_split were tuned by hand. All tests were completed with default settings other than the specified parameter.

CRITERION	AVERAGE PRECISION	AVERAGE RECALL
ENTROPY	0.44485	0.42350
GINI	0.42696	0.45450

Table 9 Entropy vs Gini criterion

Entropy and Gini criterion give trade-offs between precision and recall where using Gini results in a reduction in precision but an almost equal level of improvement in recall. Neither is better than the other, the selection of the criterion will depend which of recall or precision is more important.

For this assignment, I have selected recall as more important. Therefore, Gini was selected as the criterion for the classifier. See next section “Analysis Validation and Performance” for further discussion on recall and precision.

MIN_SAMPLES_SPLIT	AVERAGE PRECISION	AVERAGE RECALL
2	0.42142	0.45050
3	0.45587	0.45200
4	0.45304	0.45100
5	0.43442	0.41900
6	0.44104	0.42450

Table 10 min_samples_split

As min_samples_split increases the precision increases but the average recall remains about the same. The optimal setting is 4 where precision and recall start decreasing with any higher value.

Accuracy	Precision	Recall	F1	F2
0.84357	0.45192	0.44650	0.44920	0.44757

Table 11 Final metrics with C = Gina and min_samples_split = 4

Feature importance:

- exercised_stock_options (0.300090)
- expenses (0.264095)
- shared_receipt_with_poi (0.239094)
- other (0.196720)

Analysis Validation and Performance

What is validation, and what's a classic mistake you can make if you do it wrong

In supervised machine learning, validation is the process of testing the trained model against another set of data with known labels. The predicted label of the test data is compared against the actual labels to see how the model performs and to see if it generalises well. A classic mistake in validation is overfitting, where the model performs well on the training and test data sets but poorly on other data sets. To mitigate this risk I used cross-validation so that the model was validated against a diverse range of test data sets.

The data was split into training and testing sets using stratified cross validation shuffle split with n_iter=1000 and test_size=0.1. The model was trained on the training data and then tested on the test data, all metrics were then averaged across 1000 iterations.

The decision tree model returned a precision of 0.451 and a recall of 0.447. These values are acceptable given the small sample and the low number of persons of interest. In this case, the accuracy metric is not a good indication of performance. 88% of people in the dataset are not persons of interest so I would expect all models to have a high level of accuracy by simply predicting non persons of interest correctly.

A higher recall value means that the likelihood of persons of interests in the dataset are identifier is higher, and a higher precision value means that the persons of interests identified are in fact persons of interest. My goal for this model was to identify all possible persons of interest. I was prepared to deal with a large number of false positives to ensure that all people involved in the fraud are identified. Therefore, I favour higher recall over precision.