Identifying Person of Interest Related to the Enron Scandal

1. Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those? [relevant rubric items: “data exploration”, “outlier investigation”]

Project Description

This project would build a supervised machine learning model that identifies people who were involved in Enron fraud scandal in Oct, 2001, using publicized financial and email dataset of each 145 top executives of the Enron during that period. This dataset is consist of 145 records, each record is mapped to each person, with 20 features related to financial and email information.

The classes of the dataset, Person of Interest (PoI), were labeled True for people who took legal responsibility for the scandal, and others were labeled False. There are 18 persons are labeled True ‘poi’, and 128 others are labeled False.

Data Exploration & Handling Outliers and Anomalies

To explore patterns and anomalies of data, I drew boxplots for each features by ‘poi’ class. I could discover that there are some data points of extremely large value for most of features as you can see from boxplots below. It turned out that those data points were grand total of entire values accidently included in the dataset, mapped to “Total” index, and those points were removed. But other outliers were retained since those are regarded as making sense.

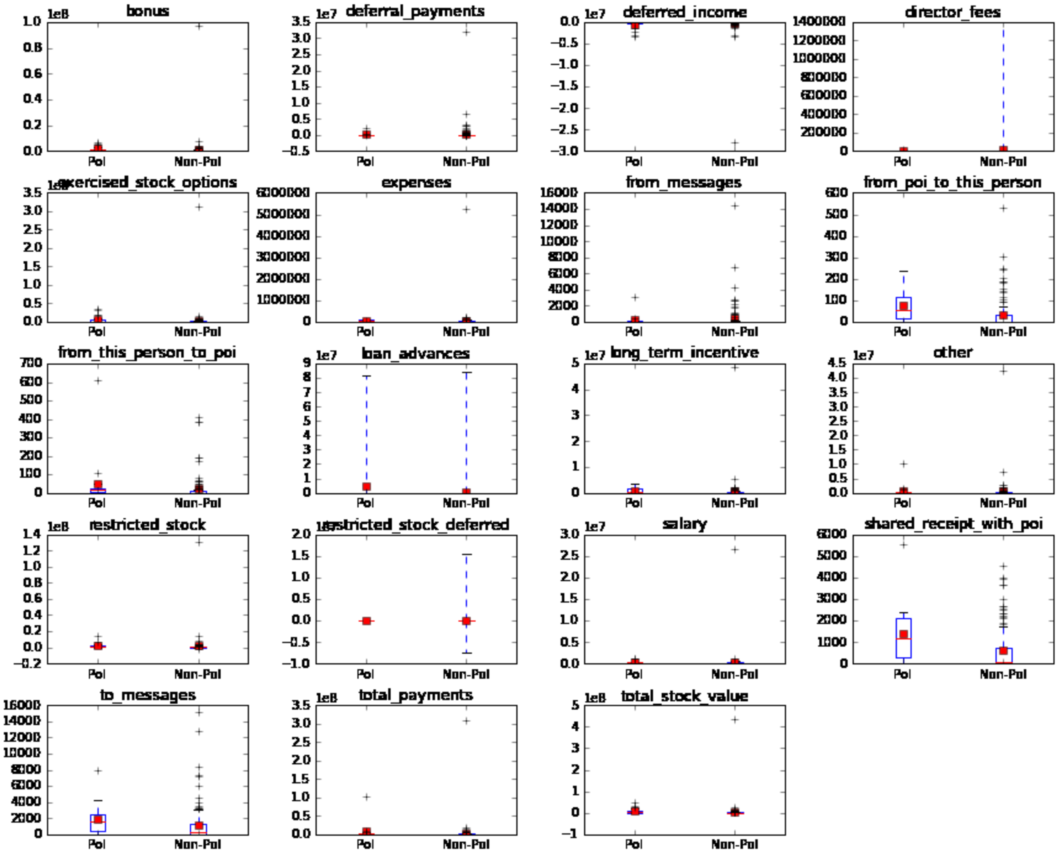


Figure 1 Distribution of values each features by class (Original dataset)

Outliers Identification

Original Dataset	After "Total" record removed
`*** total_payments`	`*** total_payments`
` max : TOTAL, False, 309886585`	` max : LAY KENNETH L, True, 103559793`
` min : CHAN RONNIE, False, NaN`	` min : CHAN RONNIE, False, NaN`
`*** expenses`	`*** expenses`
` max : TOTAL, False, 5235198`	` max : MCCLELLAN GEORGE, False, 228763`
` min : BAZELIDES PHILIP J, False, NaN`	` min : BAZELIDES PHILIP J, False, NaN`
`*** from_messages`	`*** from_messages`
` max : KAMINSKI WINCENTY J, False, 14368`	` max : KAMINSKI WINCENTY J, False, 14368`
` min : BADUM JAMES P, False, NaN`	` min : BADUM JAMES P, False, NaN`
`*** long_term_incentive`	`*** long_term_incentive`
` max : TOTAL, False, 48521928`	` max : MARTIN AMANDA K, False, 5145434`
` min : BADUM JAMES P, False, NaN`	` min : BADUM JAMES P, False, NaN`
`*** restricted_stock`	`*** restricted_stock`
` max : TOTAL, False, 130322299`	` max : LAY KENNETH L, True, 14761694`
` min : BHATNAGAR SANJAY, False, -2604490`	` min : BHATNAGAR SANJAY, False, -2604490`
`*** salary`	`*** salary`
` max : TOTAL, False, 26704229`	` max : SKILLING JEFFREY K, True, 1111258`
` min : BADUM JAMES P, False, NaN`	` min : BADUM JAMES P, False, NaN`
`*** to_messages`	`*** to_messages`
` max : SHAPIRO RICHARD S, False, 15149`	` max : SHAPIRO RICHARD S, False, 15149`
` min : BADUM JAMES P, False, NaN`	` min : BADUM JAMES P, False, NaN`
`*** total_stock_value`	`*** total_stock_value`
` max : TOTAL, False, 434509511`	` max : LAY KENNETH L, True, 49110078`

min : BELFER ROBERT, False, -44093 *** loan_advances max : TOTAL, False, 83925000 min : ALLEN PHILLIP K, False, NaN	min : BELFER ROBERT, False, -44093 *** loan_advances max : LAY KENNETH L, True, 81525000 min : ALLEN PHILLIP K, False, NaN

Dealing with Missing Values

After missing value exploration described below, four features— 'deferral_payments', 'director_fees', 'loan_advances', and 'restricted_stock_deferred' —that contain missing values more that 70% over the whole record for both True and False 'poi' label were excluded for building model. In addition, data related to mails – 'from_messages', 'from_poi_to_this_person', 'from_this_person_to_poi', 'shared_receipt_with_poi', 'to_messages'– were missing 44 and 56 records in 'poi' class and 'non-poi' class respectively. These missing values were replaced with median of non-missing values of corresponding 'poi' label for each feature, rather than removing or substituting with zero. This decision was made under an assumption that usually most of employees of the company would use emails actively.

Missing Values Exploration

Number of PoI : 18, Number of non-PoI: 128

*: features contains missing values more than 70% for both 'poi' and 'non-poi'

**: features related to mail

label Feature	non_poi_ missing (count)	non_poi_ missing_ prop	poi_ missing (count)	poi_ missing_ prop
*director_fees	111	87%	18	100%
*restricted_stock_deferred	110	86%	18	100%
*loan_advances	125	98%	17	94%
*deferral_payments	94	73%	13	72%
deferred_income	90	70%	7	39%
exercised_stock_options	38	30%	6	33%
long_term_incentive	74	58%	6	33%
**from_messages	56	44%	4	22%
**from_poi_to_this_person	56	44%	4	22%
**from_this_person_to_poi	56	44%	4	22%
**shared_receipt_with_poi	56	44%	4	22%
**to_messages	56	44%	4	22%
bonus	62	48%	2	11%
restricted_stock	35	27%	1	6%

salary	50	39%	1	6%
expenses	51	40%	0	0%
other	53	41%	0	0%
poi	0	0%	0	0%
total_payments	21	16%	0	0%
total_stock_value	20	16%	0	0%

2. What features did you end up using in your POI identifier, and what selection process did you use to pick them? Did you have to do any scaling? Why or why not? As part of the assignment, you should attempt to engineer your own feature that does not come ready-made in the dataset -- explain what feature you tried to make, and the rationale behind it. (You do not necessarily have to use it in the final analysis, only engineer and test it.) In your feature selection step, if you used an algorithm like a decision tree, please also give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values. [relevant rubric items: “create new features”, “properly scale features”, “intelligently select feature”]

Feature Selection

Adding New Engineered Features

For better performance of classifier, I created two features. Proportion of mails including ‘poi’ are computed for each ‘sent’ or ‘received’ mails, expecting that it might be more sensitive to detect ‘poi’ besides its absolute amount of mails.

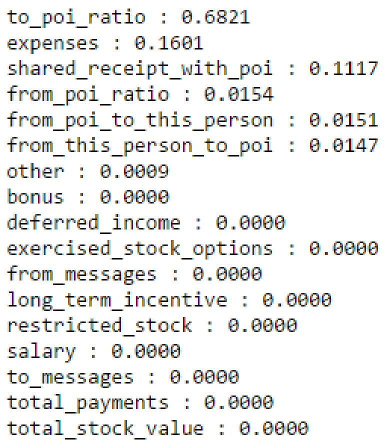
1. from_poi_ratio = from_this_person_to_poi/ from_messages
2. received_poi_ratio = from_poi_to_this_person / to_messages

Computing Importance of Features and Select Best Features

To select most informative features among entire features, computed importance of each feature using gini index.

Features with 0 importance — bonus, deferred_income, exercised_stock_options, from_messages, long_term_incentive, restricted_stock, salary, to_messages, total_payments, total_stock_value—were also removed. As a result, seven features were selected to build prediction models. On top of that, the number of features were tuned through grid search in a range of 2 and 7 to get the best performance for each algorithm.

Table 1 Importances of features



```
to_poi_ratio : 0.6821
expenses : 0.1601
shared_receipt_with_poi : 0.1117
from_poi_ratio : 0.0154
from_poi_to_this_person : 0.0151
from_this_person_to_poi : 0.0147
other : 0.0009
bonus : 0.0000
deferred_income : 0.0000
exercised_stock_options : 0.0000
from_messages : 0.0000
long_term_incentive : 0.0000
restricted_stock : 0.0000
salary : 0.0000
to_messages : 0.0000
total_payments : 0.0000
total_stock_value : 0.0000
```

3. What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms? [relevant rubric item: “pick an algorithm”]

Training Classifiers and Selecting Best Model

I selected Random Tree algorithm based on its best performance over Naïve Bayes, Decision Tree and Adaboost algorithm with optimized tune and feature selection settings. Performance was measured by a customized evaluation metric, which is based on harmonic average of recall and precision of True ‘poi’ label. Additionally, this score is weighted in case both recall and precision are over .3 as described below to easily detect algorithms that meet the goal of this project.

$$\text{Performance Score} = \frac{2}{\dfrac{1}{recall('poi' = True)} + \dfrac{1}{precision('poi' = True)}} * weight$$

weight = 100, if recall('poi'=True)> .3 and precision('poi'=True)> .3

= 1, otherwise

Table 2 Performance of Algorithms

	Precision (mean)	Recall (mean)	Performance Score
Naïve Bayes	0.05	0.07	0.06

> PCA	0.04	0.06	0.17
DecisionTree	0.56	0.66	82.61
> PCA	0.37	0.42	64.58
Random Forest	0.72	0.67	84.24
> PCA	0.39	0.44	65.35
Ada Boost	0.45	0.54	62.63
>PCA	0.44	0.38	63.33

4. What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well? How did you tune the parameters of your particular algorithm? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly explain how you would have done it for the model that was not your final choice or a different model that does utilize parameter tuning, e.g. a decision tree classifier). [relevant rubric item: "tune the algorithm"]

Parameters Tuning

Parameters of an algorithm need to be optimized for best performance, or our models would be easily biased or overfitted. I went through grid search for three influential parameters of the Random Forest model— class weight (None, balanced), max numbers of features ('auto', 'log2', None), and minimum size for to split sample (2~10) — to find the best combination of parameters of a model that performs best. Other parameters were left as default. As a result, I could get the best decision tree classifier with parameters as below.

- class_weight='balanced'
- max_features='auto'
- min_samples_split=5

5. What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis? [relevant rubric item: "validation strategy"]

Model Validation

We need to validate our classification models through cross validation. Since, classifiers can be easily overfitted or biased depending on its training data. For cross validation, first, we need to split original dataset into train dataset and test dataset. Then trained classification models with training data and validate with test. Which method enables us to check the model is not overfitted: making sure that model is generalized to work well with independent dataset.

Also, before splitting data, rows of the data would better to be shuffled and test would better to be implemented several times with randomly selected subsets respectively to minimize bias and variance of subset caused by unbalanced distribution of class in the dataset.

6. Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something human-understandable about your algorithm's performance. [relevant rubric item: "usage of evaluation metrics"]

Evaluation

In this project, it was more important to detect 'poi' class(True) rather than 'non-poi' class(False). So, Recall and precision of 'poi' class (True) are used to evaluate classifiers in this project. Recall is proportion of correct positive classifications over entire positive records. This metric measures how much proportion the model detects 'poi' class over entire 'poi's in this project. As opposed to recall, precision— proportion of correct positive classifications over the number of total positive classifications— measures correctness of True 'poi' classifications in this project.

Since both correctness and detectability are important in this project, I used average of recall and precision. On top of that the metric get weighted if both its recall and precision are over .3 to meet the goal of the project.

With the selected classifier, I've got a result of precision as .7240 and recall as .64 on original dataset. (performance score: 84.24)

$$\text{Performance Score } [0\sim100] = \frac{2}{\frac{1}{recall('poi' = True)} + \frac{1}{precision('poi' = True)}} * weight$$

weight = 100, if recall('poi'=True)> .3 and precision('poi'=True)> .3

= 1, otherwise