Identifying Persons of Interest in the Enron Fraud Case

Source Code - Github: https://github.com/rakeshdhote/EnronFraudInvestigation

Introduction

Enron was one of the tenth largest American energy, commodities, and services company in 2000. The multi-billion-dollar company collapsed into bankruptcy by 2002 due to widespread corporate fraud and corruption. Thousands of people lost their jobs, and some of them indicted. In the resulting federal investigation, confidential information such as financial and email communication data referred as the *Enron Corpus* is released for public records.

The aim of this project is to identify (classify) persons of interest (POIs) and predict culpable persons using features from the *Enron Corpus* and the labeled data for POIs who were indicted, settled without admitting guilt, or testified in exchange for immunity. The lessons learned during the project would help us automatically classify POIs and can be used in fraud detection and counter terrorism.

Literature Review

The *Enron Corpus* is one of the biggest public dataset of email messages. It has been studied and investigated under different contexts. Klimt and Young [1] introduced the cleaned dataset to public. The paper analyzed suitability of the corpus for exploring how to classify messages as organized by a human. Later, they [2] implemented an automatic classification of email messages using support vector machine under various conditions.

Keila and Skillicorn [3] investigated the structure of corpus using singular value decomposition and semi-discrete decomposition. They analyzed word frequency profiles and stated that the messages fell into two distinct groups whose extrema are characterized by short messages and rare words versus long messages and common words. They inferred that the alleged criminal activity uses slightly distinctive words.

The corpus is also famous among researchers in social and network analysis. Huang and Zeng [4] used the corpus to detect anomaly problems using a graph-theoretic link prediction

approach in order to monitor traffic over various communication channels. Shetty and Abidi [5] performed a statistical analysis of the corpus and derived a social network from the email communication with a predefined threshold.

Carvalho and Cohen [6] used the *Enron Corpus* and developed a recipient recommendation system, i.e. suggesting who recipients of a message might be, while the message is being composed, given its current contents and its previously-specified recipients. Such recommender is particularly useful in large corporation for identifying people working on similar topics, or project, or to find people with appropriate expertise or skills. Peterson et al. [7] used the corpus to focus on email formality and explore the factors that could affect the sender's choice of formality. They studied how formality is affected by social distance, relative power, and the weight of imposition.

Though the corpus is investigated under different contexts, there has been scarce investigation in regards to how to identify (classify) a person as a POI or non-POI based on their email communications and financial data. This capstone project focuses on investigating this task.

Dataset

The *Enron Corpus* is downloaded from the CMU data dump [8] and the email and financial data are obtained from the Github repo [9]. The dataset consists of 146 user records each with 14 financial features, six email features, and one labeled feature (POI). Financial data include features like salary and bonus while the email features include a number of messages written/received and to whom/from. The features in a corpus are listed below:

`bonus, deferral_payments, deferred_income, director_fees, email_address, exercised_stock_options, expenses, from_messages, from_poi_to_this_person, from_this_person_to_poi, loan_advances, long_term_incentive, other, poi, restricted_stock, restricted_stock_deferred, salary, shared_receipt_with_poi, to_messages, total_payments, total_stock_value.`

Exploratory Data Analysis

A quick look at the data revealed 18 POI's in the dataset. Some of these were high ranked officials in the Enron Corporation.

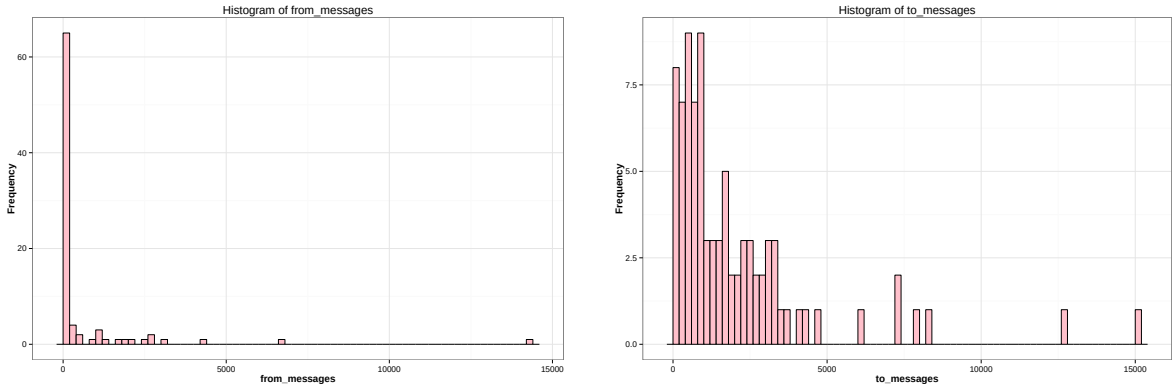
```
BELDEN TIMOTHY N     BOWEN JR RAYMOND M   CALGER CHRISTOPHER F
```

CAUSEY RICHARD A	COLWELL WESLEY	DELAINEY DAVID W
FASTOW ANDREW S	GLISAN JR BEN F	HANNON KEVIN P
HIRKO JOSEPH	KOENIG MARK E	KOPPER MICHAEL J
LAY KENNETH L	RICE KENNETH D	RIEKER PAULA H
SHELBY REX	SKILLING JEFFREY K	YEAGER F SCOTT

Now, let's have a closer look at the data. Figure 1 shows histograms of two features: from_messages and to_messages. There are few people with large email to or from communications. So, is it a good idea to remove this outliers? We need to hold on to this thought till we explore the data in more detail.

Figure 2 shows histograms for bonus, expenses, total_stock_value, total_payment, long_term_incentive, and exercised_stock_options. These histograms also have a right skewed long tail distribution with clear outliers.

Now, let's look at the scatter plot of fraction_from_poi vs. fraction_to_poi and salary vs. bonus for POIs (1) and non-POIs (0) in Fig. 3(b). It was expected that the POIs feature behavior would be similar with larger (fraction of) email communications and closely clustered together. However, there are no clear clusters of the POIs and NonPOIs. Hence we cannot remove the outliers. In such case, we can resort to the machine learning models for predicting POIs as presented in the following sections.



(a) (b)

Figure 1: Histograms of from_messages and to_messages features.

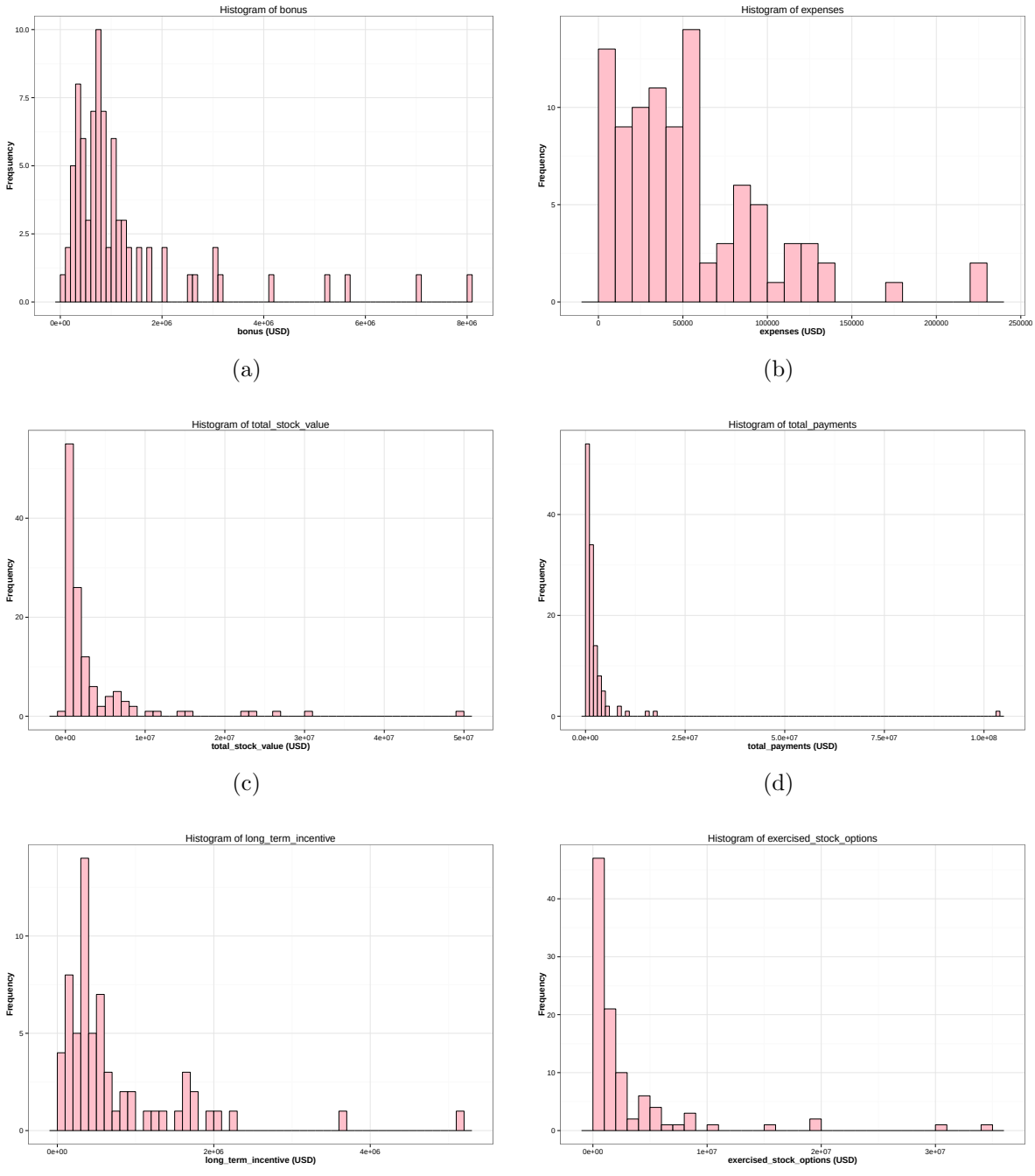
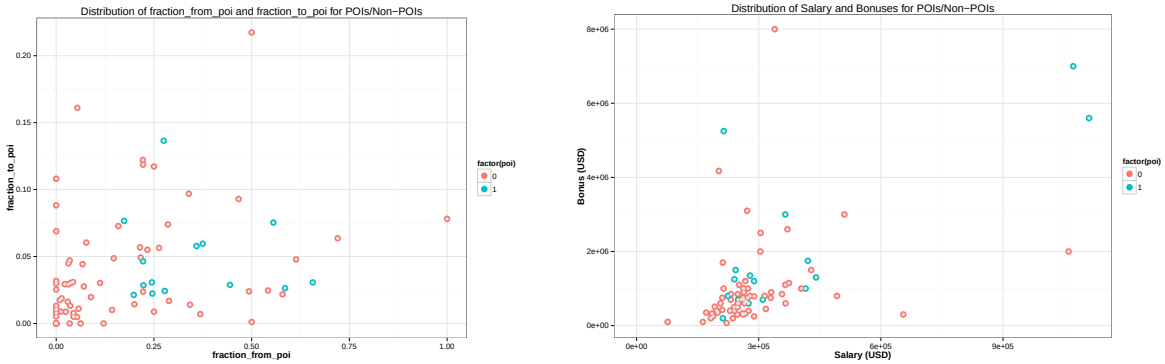


Figure 2: Histograms of various features in the dataset.



(a) (b)

Figure 3: Scatter plots of from_messages Vs. to_messages and Salary Vs. Bonus.

Approach

The objective of this project is to classify a person as POI or non-POI. Overall, the process involves data cleaning and transformation, selecting appropriate features and classification algorithm and validation. Depending on chosen performance scores, the process can be repeated by starting with a new set of features. The steps are schematically summarized in Fig. 4. The details for each step are described in the following subsections.

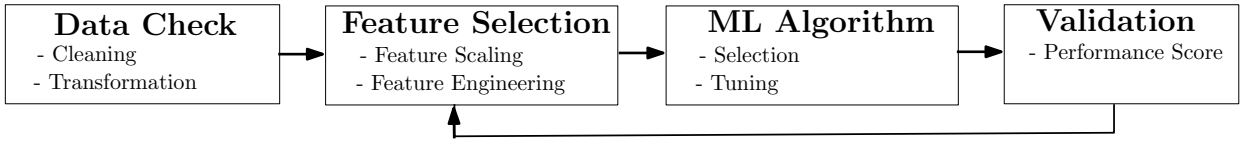


Figure 4: Schematic pipeline for classifying POI and non-POI persons in the Enron Fraud case

The modeling and analysis is conducted using Python and Scikit Learn (*SKLearn*) package. The source code can be fetched from the Github at
https://github.com/rakeshdhote/EnronFraudInvestigation

Step 1: Data Check, Cleaning and Transformation

A first step in the model development is checking the data validity and data cleaning. A quick look at the data indicates that there are two clear outliers in the data: TOTAL and THE TRAVEL AGENCY IN THE PARK. The first outlier represents an aggregate of the dataset, while the second one seems to be a random entry. Both outliers are removed during further analysis. There are few outliers with very high bonuses. These persons are not removed as they are POIs.

Additional data check steps such as the consistency of the data format and transformation have been conducted.

Step 2: Feature Selection, Scaling and Engineering

The next important step in a model development is the selection of features. Among 21 features in the dataset, it is important to select appropriate features for further investigation. Using SkLearn's *SelectKBest* algorithm, I selected the following 11 features for the analysis:

`poi, bonus, deferred_income, exercised_stock_options, fraction_from_poi, long_term_incentive, poi_ratio, restricted_stock, salary, total_payments, total_stock_value`

Some of these features have wide ranges. As an example *bonus* ranges from $70000 to $97343619 while *fraction_from_poi* ranges from 0 to 1. It is important to scale these features before introducing them in the model. SkLearn's *MinMaxScaler* is used for the feature scaling.

In addition to the above features, following three new features are engineered under the premise that POI's contact each other more frequently than non-POIs. The newly introduced features are:

- *poi_ratio*: It is a ratio defined as (from_poi_to_this_person + from_this_person_to_poi + shared_receipt_with_poi)/(from_messages + to_messages).
- *fraction_from_poi*: It is a ratio defined as from_this_person_to_poi/from_messages.
- *fraction_to_poi*: It is a ratio defined as from_poi_to_this_person/to_messages.

The above features are introduced in the model, and performance is evaluated. In the later stages, the features are hand picked for improving the model performance.

Step 3: Algorithm Selection and Performance Tuning

As the ratio of labeled data POI:non-POI is 18:126 is imbalanced, it is advisable to use SkLearn's *StratifiedKFold* for maintaining the ratio in a group during a cross validation study. In this small dataset, choosing a fold size is important in order to keep enough POI labels. SkLearn's *Pipeline* feature is leveraged for chain transformations and estimators for better flexibility. A *GridSearchCV* is utilized to conduct an exhaustive search over specified parameter values for an estimator. In particular, the cross validation is used to avoid over or under fitting issues in the data.

In the initial pass, automatic tuning of parameters is conducted. On obtaining a better parameter space, local minima is achieved by fine tuning parameters.

Step 4: Performance Evaluation

The accuracy measure is generally used for classification problems. However, in this highly imbalanced class (18 POI and 126 Non-POI) problem, accuracy is not a true performance measure. Because of the skewed nature of the labeled data, identifying non-POIs would always yield high-accuracy scores. Hence in the subsequent analysis, accuracy is not used for the performance evaluation.

Instead, to evaluate performance of each algorithm, the precision, recall and F1 scores are calculated and compared for different classifiers. The precision refers to a fraction of retrieved instances that are relevant, while recall refers to a relevant instances that are retrieved, and the F1 score refers to a weighted average of the precision and recall.

The modeling procedure is iterated for each algorithm to improve the performance. Based on the best performance measure, a final algorithm is selected.

Results

To recap, the objective of this project is to classify whether a person in the Enron Corpus is a POI or non-POI. Different classifiers have been considered using *StratifiedKFold*, *PipeLine*, and *GridSearchCV* (cross validation) for combination of algorithm parameters. Tuning algorithm parameters made a significant impact on performance scores. Table 1 summarizes average performance scores for different classification algorithms.

Table 1: Classifier's Average Performance Scores

Classifier	Precision	Recall	F1	Accuracy
Decision Tree	0.5238	0.5667	0.5370	0.8489
Logistic Regression	0.3333	0.0667	0.1111	0.8255
Adaptive Boost	0.7833	0.5833	0.6405	0.8957
Random Forest	0.2778	0.1500	0.1905	0.8255
k Nearest Neighbor	0.2778	0.2167	0.2333	0.7668
Naive Bayes	0.3333	0.0667	0.1111	0.8489
Gradient Boost	0.5556	0.4333	0.4815	0.8489

Higher precision value indicates the presence of a larger number of classified true positives (POIs) among total true and false positives. Similarly, higher value of recall indicates the presence of a larger number of classified true positives (POIs) among total true positives and false negatives. In other words, higher precision indicates better quality results, while recall indicates better quantity (complete) results. The F1 score is a weighted measure of precision and recall.

The classifiers such as Naive Bayes, Logistic regression and k nearest neighbor (kNN) algorithms performed poorly with low precision, recall and F1 measures for different parameter combinations. On the other hand, the decision tree performed relatively better. Surprisingly, the random forest has poor algorithmic performance than other ensemble methods. The Gradient Boost provides better performance score, while adaboost (adaptive boosting) provided the best scores among all classifiers. The final algorithmic parameters of adaptive boost classifier are

```
AdaBoostClassifier(algorithm='SAMME.R', learning_rate=1.0, n_estimators=25,
                   random_state=None).
```

The model (adaptive boosting algorithm) predicts the POIs with 78% confidence (precision) and 64% weighted score (F1). These numbers are quite good for imbalanced class dataset in hand. However, there is still room for improvement. With relative consistent performance among different classifiers, the results are promising and confident.

Final Thoughts

Following are the additional possible avenues which can improve performance score in the future.

- Generate additional features considering time line history and time stamp of email between people under the premise that POIs exchange communication more frequently during non-business hours.
- Parse financial data from email communications from/to POIs.
- Use more sophisticated algorithms (ex. neural network) for modeling.
- Perform graph analysis to obtain insights about patterns of communication among POIs.
- Conduct topic modeling on email communication between POIs to identify common themes.

Conclusions

Overall, the aim of this capstone project was to classify persons of interest (POI) in the *Enron* fraud case. The email and financial data is used to for modeling and analysis. The data is cleaned, appropriate features are selected and engineered followed by scaling. Different classifiers are used to measure model performance. The classifiers are tuned to gain better

performance. The AdaBoost (adaptive boosting) ensemble classifier gave the best results for a highly imbalance labeled classification problem in hand.

The algorithmic pipeline developed during the project helped to automatically classify POI and non-POI based on their email and financial data. The methodology and code developed during this project can be extended to infer cyber-criminal activity, fraud detection and counter terrorism.

References

[1] Bryan Klimt and Yiming Yang. Introducing the Enron Corpus. In *CEAS*, 2004.

[2] Bryan Klimt and Yiming Yang. The Enron Corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, pages 217–226. Springer, 2004.

[3] Parambir S Keila and David B Skillicorn. Structure in the Enron email dataset. *Computational & Mathematical Organization Theory*, 11(3):183–199, 2005.

[4] Zan Huang and Daniel Dajun Zeng. A link prediction approach to anomalous email detection. In *SMC*, pages 1131–1136, 2006.

[5] Jitesh Shetty and Jafar Adibi. The Enron email dataset database schema and brief statistical report. *Information sciences institute technical report, University of Southern California*, 4, 2004.

[6] Vitor R Carvalho and William Cohen. Recommending recipients in the Enron email corpus. *Machine Learning*, 2007.

[7] Kelly Peterson, Matt Hohensee, and Fei Xia. Email formality in the workplace: A case study on the Enron corpus. In *Proceedings of the Workshop on Languages in Social Media*, pages 86–95. Association for Computational Linguistics, 2011.

[8] Enron Corpus. `https://www.cs.cmu.edu/~./enron/`. Accessed: 2016-02-10.

[9] Email list of POI. `https://github.com/udacity/ud120-projects/tree/master/final_project`. Accessed: 2016-02-10.