BT5152 Decision Making Technology for Business

Final Project
Identifying Fraud from Enron Email and Financial Data

Team:
Amit Prusty - A0186053E
Architha Kishore - A0185991L
Onni Niemela - A0185992J
Spatika Narayanan - A0088416X
Swetha Narayanan - A0074604J

Table of Contents

Research Objectives

Enron was an American energy and commodities company based in Houston, Texas, which was charged with financial fraud due to accounting malpractices in 2001, leading the company to file for bankruptcy soon after. The purpose of this study is to identify key individuals who committed fraud by analyzing financial and email data. These individuals are named as “people of interest”.

In this research project, we aim to offer a machine learning based tool to detect fraudulent behaviour in large organizations by studying emails and accounting data of suspected employees.

According to previous research, accounting fraud scandals often have very severe consequences. For example, if the company does not face immediate bankruptcy, the heavily declined share price will open windows for easy hostile takeovers. From the perspective of ethics, the results are far from ideal, as only a few guilty people can put thousands of innocent co-workers in jeopardy (Michael Jones: Creative Accounting, Fraud and International Accounting Scandals).

As reports conducted by BlackRock Asset Management Inc. show, investors have become more diligent about giving preference to responsible corporate governance. Regulation among corporate giants has been tightening in the past decade, and auditing has become more thorough. If companies can prevent having selfishly acting managers who commit accounting frauds, both the shareholders can save billions of dollars, and innocent employees won’t be adversely affected.

Data Set

We obtained the Enron Email Text dataset from Kaggle. We obtained the Enron Finance + Email Metadata Pkl dataset file from Udacity Projects Github Repository.

The Enron email and financial dataset is a trove of information regarding the Enron Corporation, an energy, commodities, and services company that infamously went bankrupt in December 2001 as a result of fraudulent business practices. In the aftermath of the company’s collapse, the Federal Energy Regulatory Commission released more 1.6 million emails sent and received by Enron executives in the years from 2000–2002. After numerous complaints regarding the sensitive nature of the emails, the FERC redacted a large portion of the emails, but about 0.5 million remain available to the public.

The financial data contains financial information including salary, bonus and stock options. The email data contains the emails themselves, metadata about the emails such as number received by and sent from each individual. The data was scraped from Enron Insider Pay pdf file and was made available in the form of a pkl file by Udacity.

Payment Data	**Stock Data**	**Email Data**
Salary	Exercised Stock Options	Email Content
Bonus	Restricted Stock	To Messages
Long Term Incentive	Restricted Stock Deferred	From Messages
Deferred Income	Total Stock Value	From POI to this person
Deferral Payments		From this person to POI
Loan Advances		Shared Receipt with POI
Other Payments		
Expenses		
Director Fees		
Total Payments		

Outlier Investigation and Data Cleaning

Financial + Email Metadata (tabular)

Data Imputation and handling missing values

We noticed numerous missing values (NaNs) in the dataset. According to the official pdf documentation for the financial data, values of NaN represent 0 and not unknown quantities. For the email data in turn, NaNs represent unknown information.

Hence, we replaced any NaN within financial data with zero but filled in the NaNs for the email data with the mean of the column grouped by the person of interest.

Outlier Detection

In the initial dataset that we obtained, there were 146 rows of information. At initial glance, some values seemed like outliers - such as the data point for which the total payments was at $309 million. Also, the summary of the data frame showed suspiciously huge numbers.

On checking the rows again, we noticed a row named ‘Total’ in the Person name column. The total row is what was displaying the suspect maximum numbers in the summary of the data frame. Also, there was a row for ‘Travel Agency in the Park’, which according to the documentation, was a company co-owned by Enron’s former Chairman’s sister. It is clearly not an individual that should be included in the dataset, hence we needed to remove it.

Then we checked the dataset again for other outliers. Given the relatively small dataset, outliers could actually indicate a significant fraudulent activity. Hence we needed to be careful while removing any rows in the dataset.

Email Data (textual)

Data Cleaning

The data set that we obtained from Kaggle contained the email content of all the individuals in the raw format. We needed to perform significant data wrangling to extract emails sent by each person of interest.

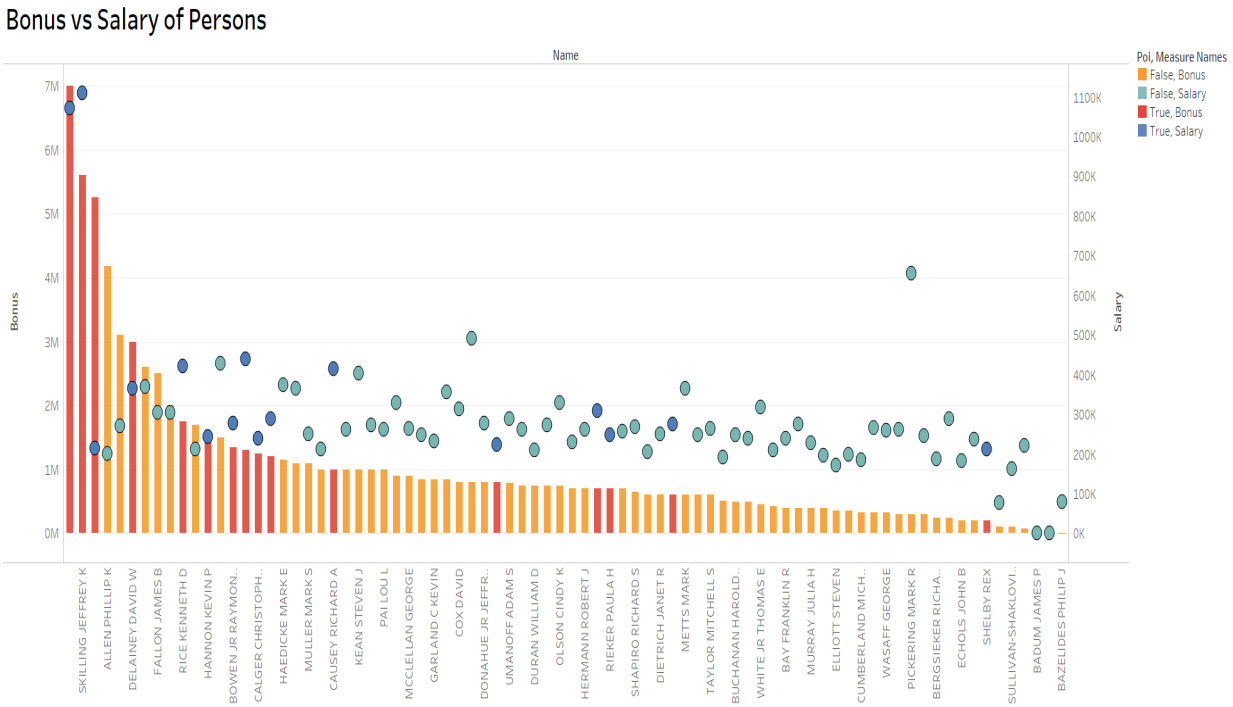
Also, since the names in the tabular data did not exactly match with the email data (“names” were actually their email addresses and folder names), we spent quite a bit of time to come up with an automated text processor that chunks the emails for each person in the tabular dataset. To do so first we assigned correct name labels to each email, filtered out every email that did not correspond to our metadata names and combined all the emails by a person into one a single documents name wise.

Data Merging

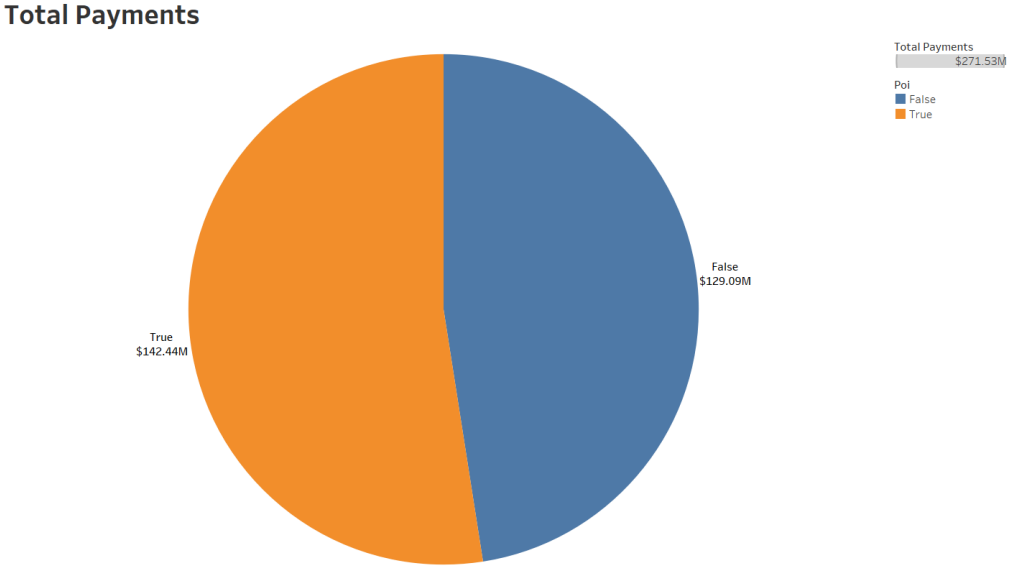
We merged the above-obtained email text data with the financial data. We obtained 126 unique rows with 17 labelled ‘Person of interest’ at the end of this process.

Data Visualization

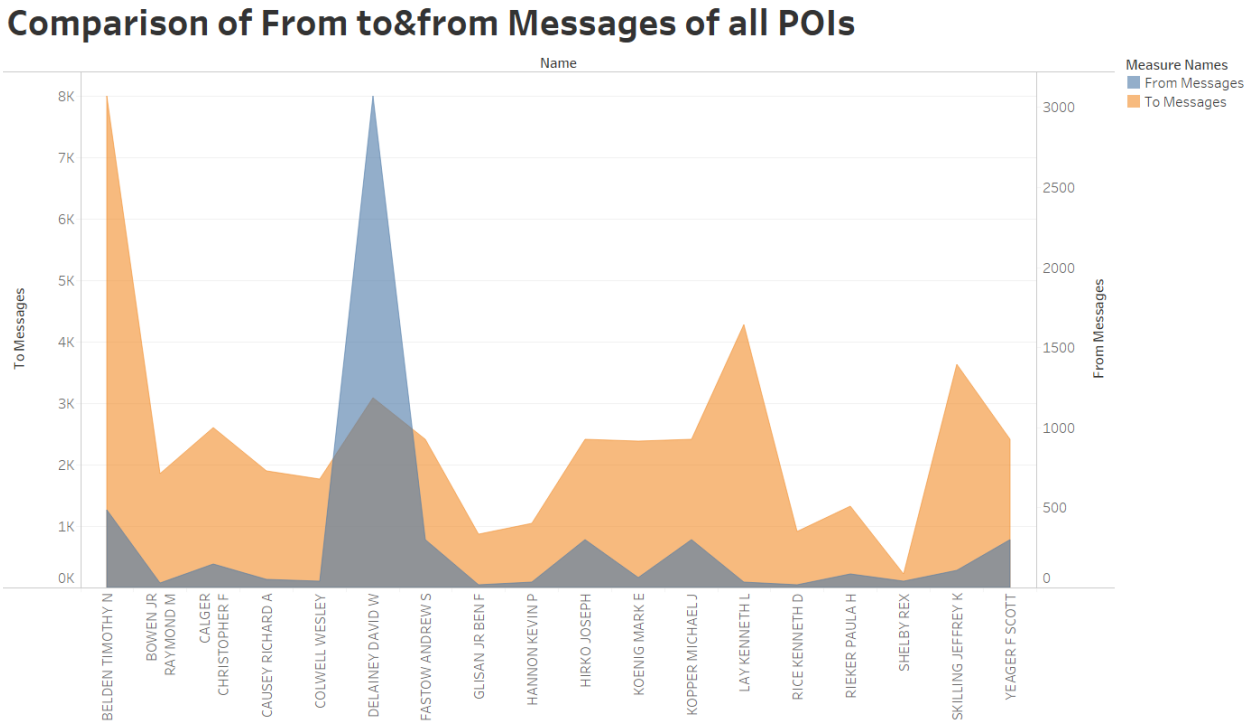
We can see below that bonuses of fraudsters are very high compared to their salary (we use this later for feature engineering). Key: Salary - Circles; Bonus-Bars.



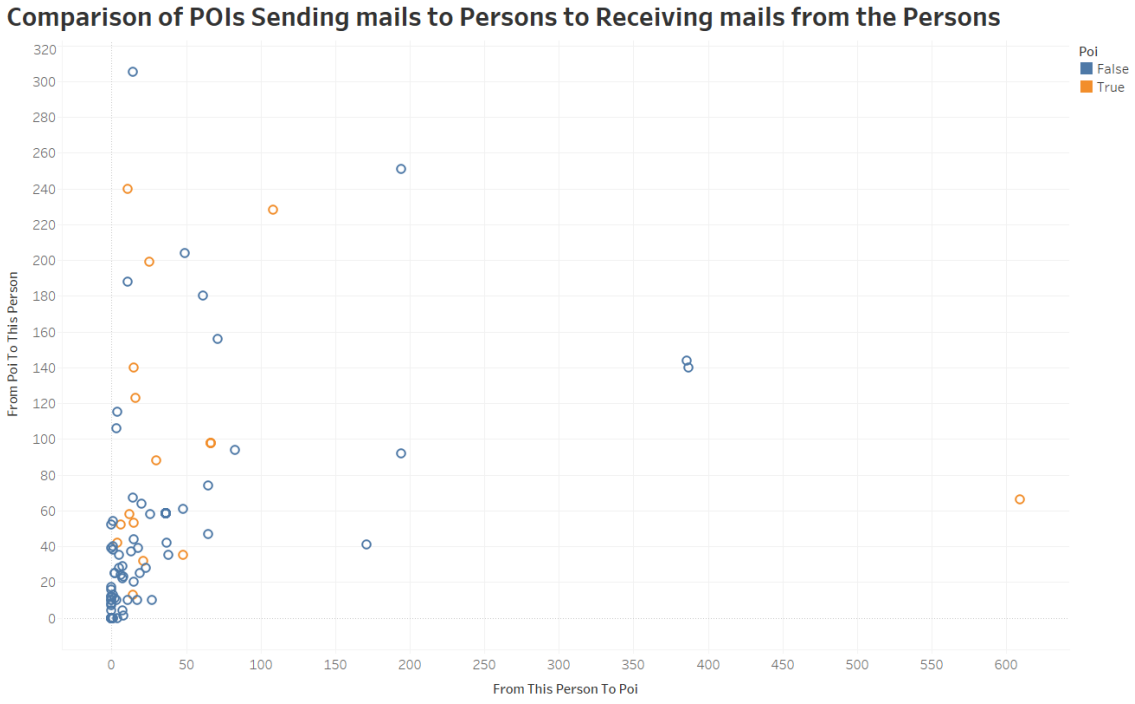
Even though there were only 18 persons of interest out of 140 in the original dataset, they took a larger share of total payments:



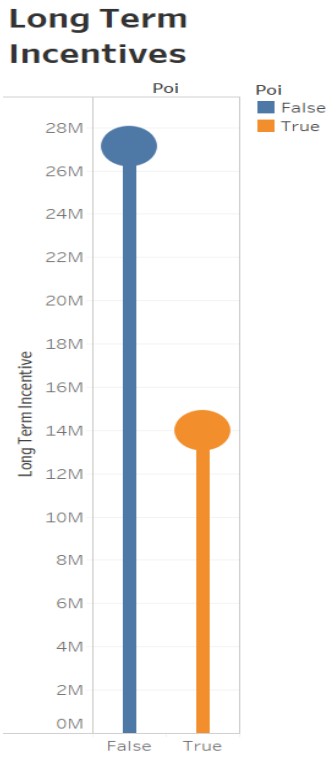
Below shows the from and to messages skewed between persons of interests, showing that they tried to contact many others, indicating possible unsavoury dealings.



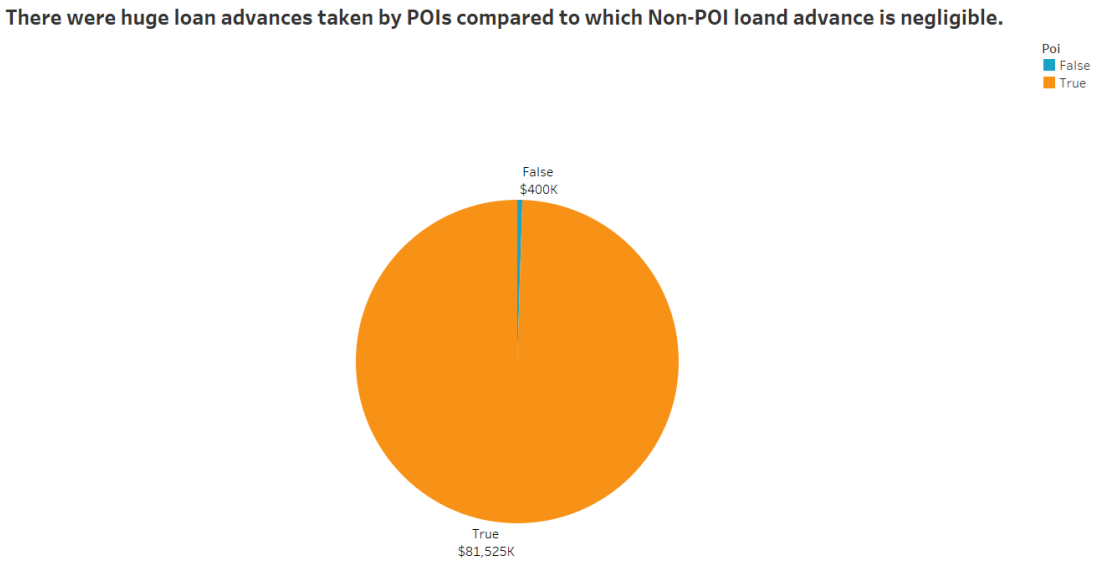
As expected the ratio of emails sent by POIs to some persons is very high compared to the emails they received from said persons:



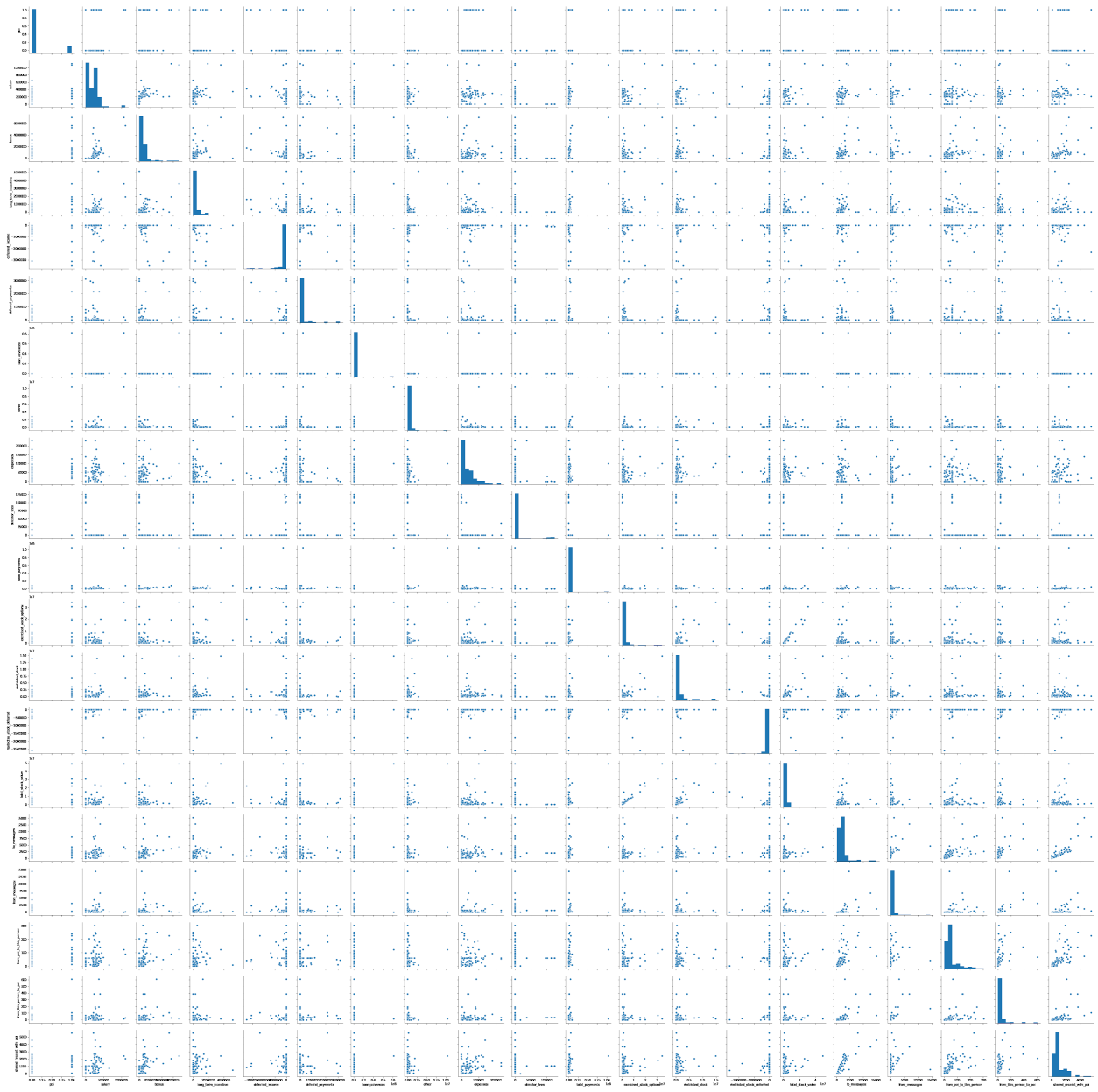
As long-term incentives of non-POIs were far better, they were less likely to commit fraud.



Huge loan advances show they were syphoning money through loans from company funds to own accounts.



We performed some initial analysis to understand the distribution and collinearity of the dataset before building the models. As we can see below some features are linearly correlated which might lead to overfitting for weaker algorithms such as rpart.



Feature Construction from Text Mining

Text Pre-Processing Steps

We followed standard preprocessing steps to clean up our email corpus: converting all words to lowercase, replacing emojis with spaces, removing numbers, stop words and punctuation. After the document was stemmed, there were certain special characters that remained that we replaced with spaces. Finally, the extra white spaces were stripped. The DocumentTermMatrix was then constructed using TF-IDF weighting.

Dictionary Approach

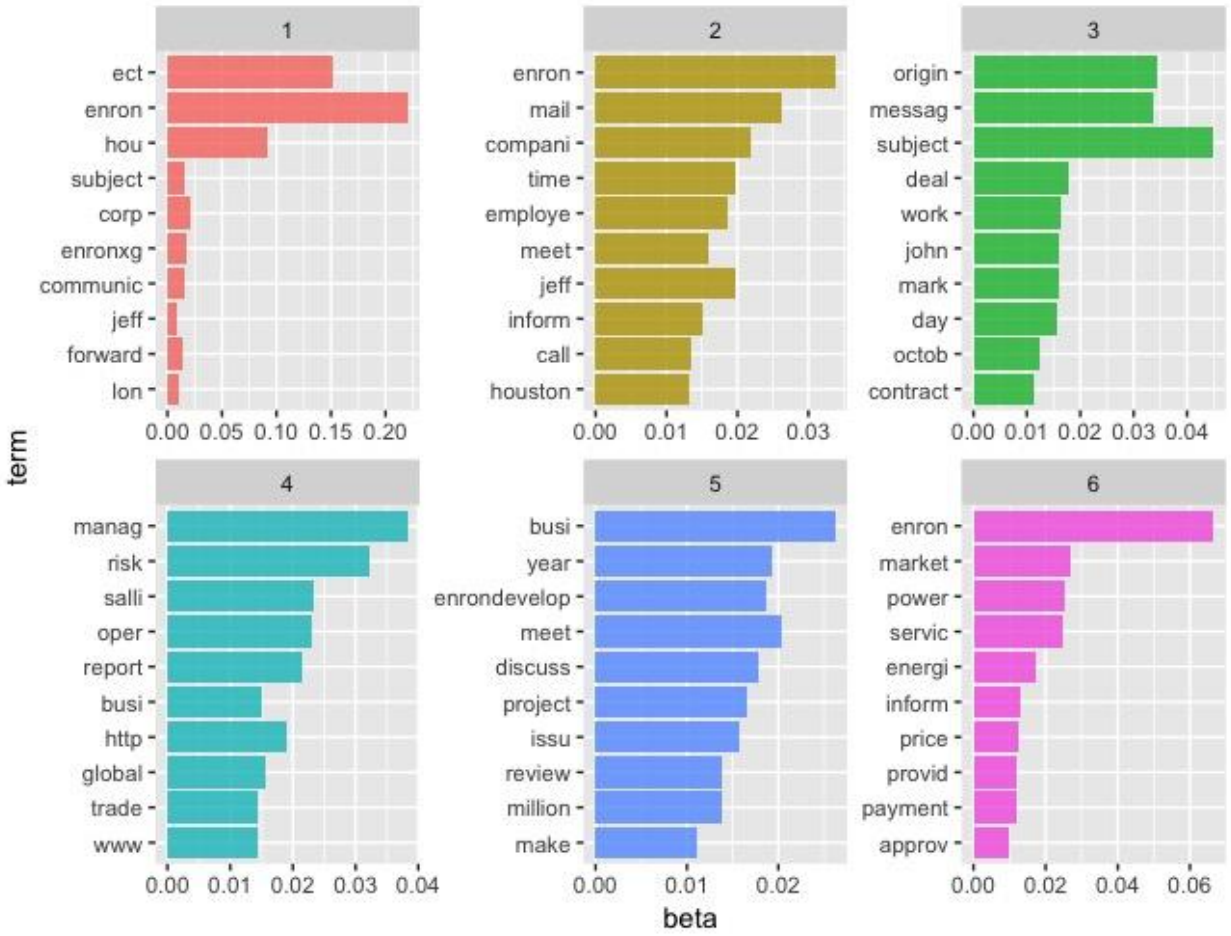
After the text pre-processing on the organized emails, we used two dictionaries from GeneralInquirer: “Negativ” and “RcEthics”, for the corresponding scores of each document in our DocumentTermMatrix. The reasoning behind this is that those who are cognizant of ethics and use many ethics-related terms in their emails are less likely to be people of interest. For the negativity score, perhaps closer to the time of the scandal, several people of interest were more aware of the worsening situation, leading to usage of a more negative tone overall.

Topic Modelling

Since the content of each document is a collection of emails from a single person, we chose to represent each by a few topics - i.e. two to six. The optimal topic model (minimum perplexity) was found to have six topics, which makes sense since perplexity decreases with the number of topics. Though it would have continued to decrease with increasing k, we found six topics to be more meaningful than, say, ten or more. This aligns with the limited number of topics that work emails are likely to contain.

The topic model used was Latent Dirichlet Allocation from the topicmodels package. The document-topic distributions obtained were used as six text-based features. The potential use of this is that documents (set of emails) of persons of interest could have a distribution over topics more indicative of fraudulent activity (higher probability for these topics).

From the figure below, we see top words in Topics 1 and 2 include “Jeff” (CEO at the time of the scandal was Jeffrey Skilling) and the company itself (“Enron” or “ECT” - Enron Capital and Trade Resources). Topic 3 seems harder to pin down to one theme, while 4 is more about global operations and managing risk. Topics 5 and 6 involve terms like “price”, “payment” and “million”. Topics 1-4 were later found to be relatively important features (refer variable importance plot from results in ensemble learning).

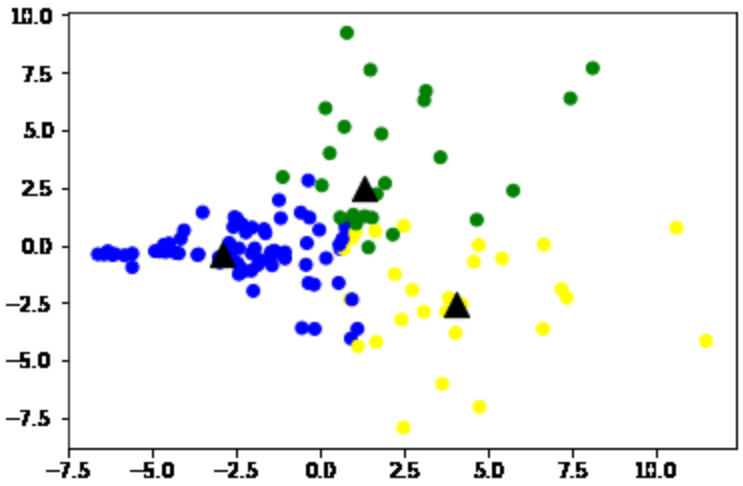


Top 10 terms per topic

Doc2Vec with KMeans Clustering

As bag of words lose a lot of meaning and document representation due to word ordering being removed, we used Doc2Vec to create text features which uses continuous bag of words, skip-gram and paragraph id to retain meaning and uses less memory because it does not keep word vectors. For Doc2Vec we kept the dimensions to 200 and window of words to be predicted as 10. We ignored any words with less than 500 frequency.

As our initial dataset is very small and features generated from the Doc2Vec model have 2000 dimensions, we did KMeans clustering to group the dimensions into 3 clusters and use these clusters as a new text feature. We then ran PCA to get 2 principal components and visualized clusters on these 2 PCAs.



Final Clusters

Latent Semantic Analysis - SVD (Singular Value Decomposition) on Tf-Idf

We created Tf-Idf matrix from the email documents ignoring any word that appears less than 5 times in all documents. Then used singular value decomposition to reduce the dimension of the matrix to size 10 as our total data points are very small.

Initial Steps

Training-Validation Split

We went with a 70:30 training-validation split on the dataset. We used createDataPartition, to ensure our training and validation sets had a balanced distribution of persons of interest.

Feature Selection

Following the principle of parsimony, we removed features that did not significantly contribute to the results. The following features had near zero variance and were removed.

- Loan Advances
- Director Fees
- Restricted Stock Deferred

Handling Imbalance

Since we are analyzing fraudulent individuals, it's understandable that 99% of individuals did not commit fraud. So, we are looking at an imbalanced dataset scenario. We used the up-sampling technique to handle this: In the dataset, we had only 17 persons of interests out of 126 persons. To improve the accuracy of the model we up-sampled the smaller class to match the

larger class size in training dataset and did not make any change to test to keep the results as accurate as possible.

Performance Metrics

Since we are analyzing fraudulent people who are much fewer in number compared to the whole dataset, we used metrics like precision and recall to measure model performance, additionally looking at overall accuracy and AUC.

Precision: It is the fraction of persons of interest predicted by the model that are truly persons of interest.
Recall: It is the fraction of the total number of persons of interest in the data that the classifier identifies.
F1: This is to take both precision and recall into account

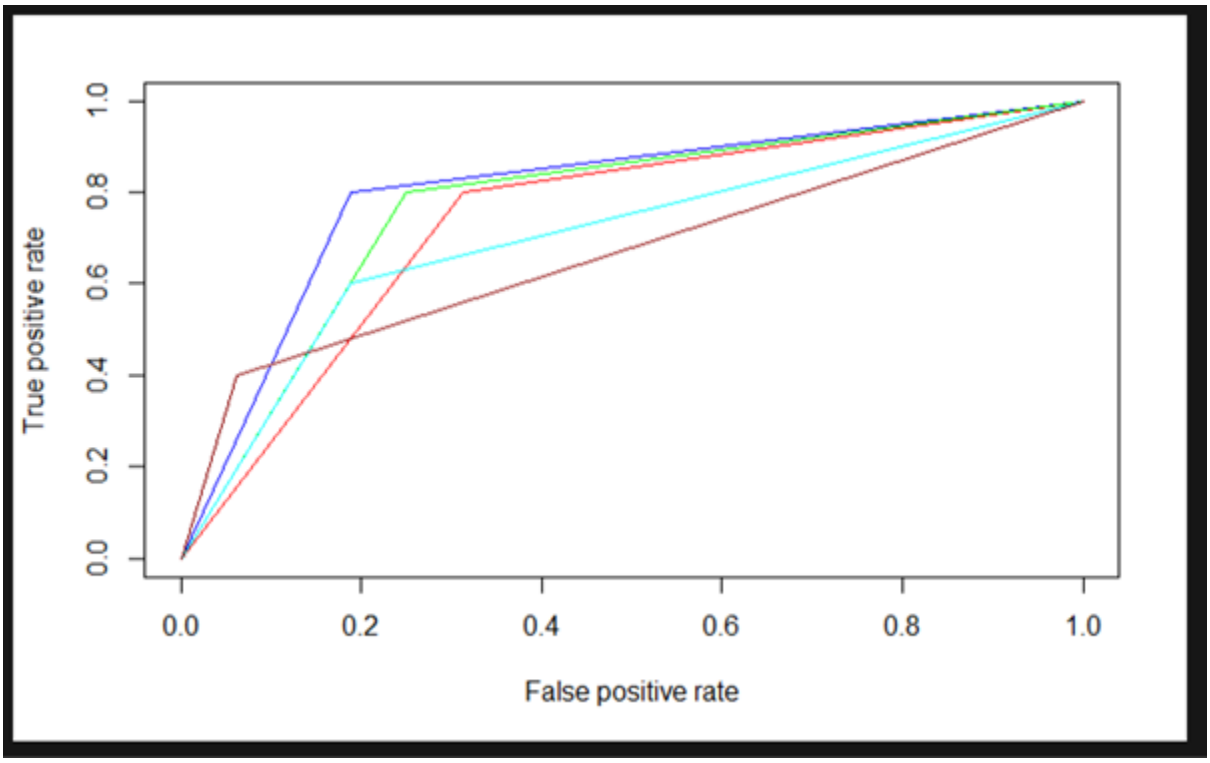
We select the best algorithm based on recall (on the validation set). This is because we believe the cost of identifying false positives (non-POI predicted as POI) is lower than not incriminating an actual POI and letting him/her go. As rpart, neural network and support vector machine are all tied with 0.8 test recall, we selected rpart, with the highest test AUC.

Baseline Results

For the benchmarking cases, we used our tabular features (minus the ones near zero variance as mentioned in feature selection), as well as our two dictionary scores, document-topic distributions for six topics, feature identifying cluster based on Doc2Vec output, and ten features from LSA-SVD. All models were cross-validated (5-fold, 2 repeats), with tuning of the relevant parameters using caret. As mentioned earlier, we used caret's up-sampling function as well to handle our imbalanced dataset.

Model	Best Tune	Training Accuracy	Training Precision	Training Recall	Test Accuracy	Test Precision	Test Recall	Test F1	Test AUC
KNN	K = 3	0.8539	0.48	1.000	0.7838	0.333	0.6	0.42	0.706
C5.0	Trials = 5, winnow = TRUE	1	1	1.0	0.84	0.4	0.4	0.4	0.65

RPart	CP = 0.1	0.8539	0.4783	0.9167	0.7568	0.333	0.8	0.471	0.775
Neural Networks	Size = 2, Decay= 0	0.898	0.61	0.91	0.6	0.19	0.6	0.29	0.6
SVM	C = 1	0.988	0.92	1.0	0.7027	0.29	0.8	0.42	0.74



The ROC curve for all the baseline models is shown above. The colour code for each of the models are: SVM - red, NN - blue, rpart - green, KNN-cyan, C50 - dark red.

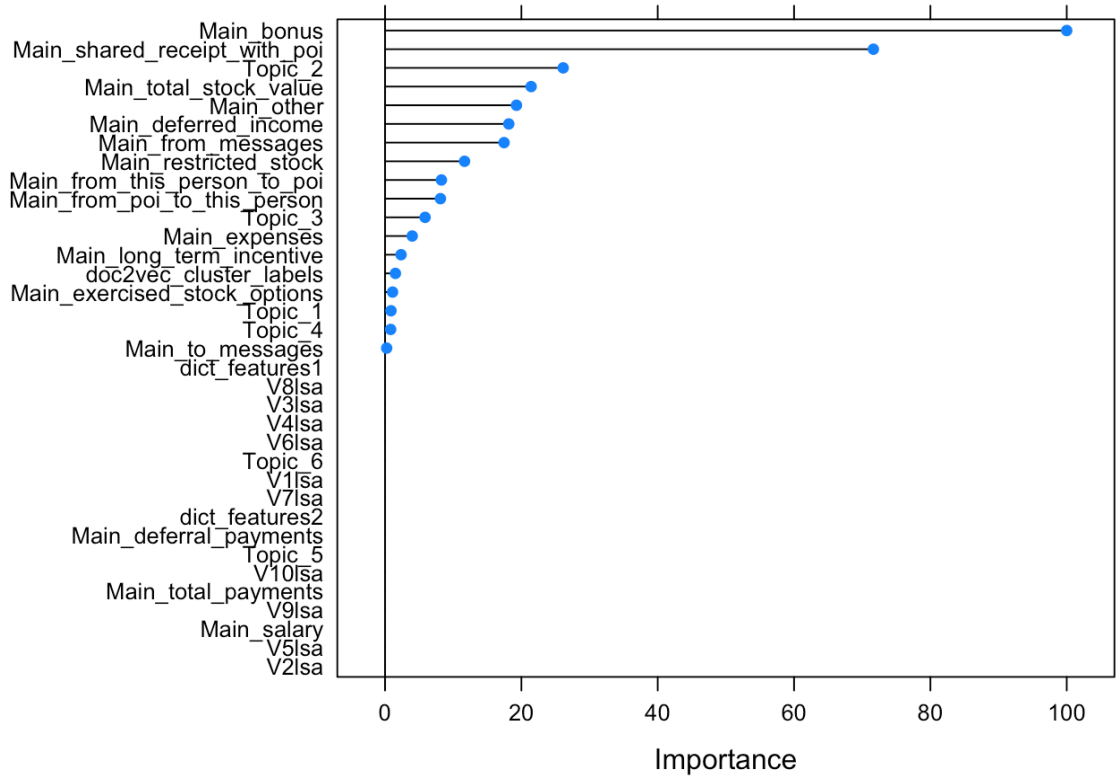
Ensemble Learning Results

For the ensemble methods, too, we used the same set of features as the baseline models, with tuning of the relevant parameters. We compare one bagging (random forest), one boosting (XGBoost) and one stacking method (k-NN, rpart and random forest).

Model Name	Best Tune	Training Accuracy	Training Precision	Training recall	Test Accuracy	Test Precision	Test Recall	Test F1	Test AUC
Random Forest with Ranger	Mtry = 11, Splitrule = Gini, Min.node.size = 1	1	1	1	0.9189	0.6	0.75	0.667	0.79
Boosting with XGBoost	nrounds= 50, max_depth= 0.4, gamma=1.5, colsample_by_tree= 0.6, min_child_weight= 5, subsample = 0.5, eta = 0.1	0.96	0.75	1	0.84	0.4	0.8	0.83	0.82
Stacking: k-NN, rpart and Random Forest	N/A	0.87	0.5	1	0.76	0.37	0.8	0.5	0.69

Baseline and Ensemble Learning Result Comparison

Using our main metric, test recall, all ensemble methods performed at least to the same level as best baseline models. However, random forest reached to 1.0 test recall, showing well improved and powerful performance. Random forest also achieved 0.92 test accuracy and 0.79 AUC, making it also our best performing model overall. When looking at AUC, ensemble models outperformed baseline models. Below, we see the variable importance plot obtained from the XGBoost results. Topics1-4, the text constructed features from topic modelling are found to be relatively useful for fraud identification.



Feature Engineering Process

We created three new features from email data (tabular) as it was apparent from initial analysis that the absolute number of emails sent by individuals was not as important as the ratio of email transactions to overall sent/received emails between persons. The three features based on this concept are:

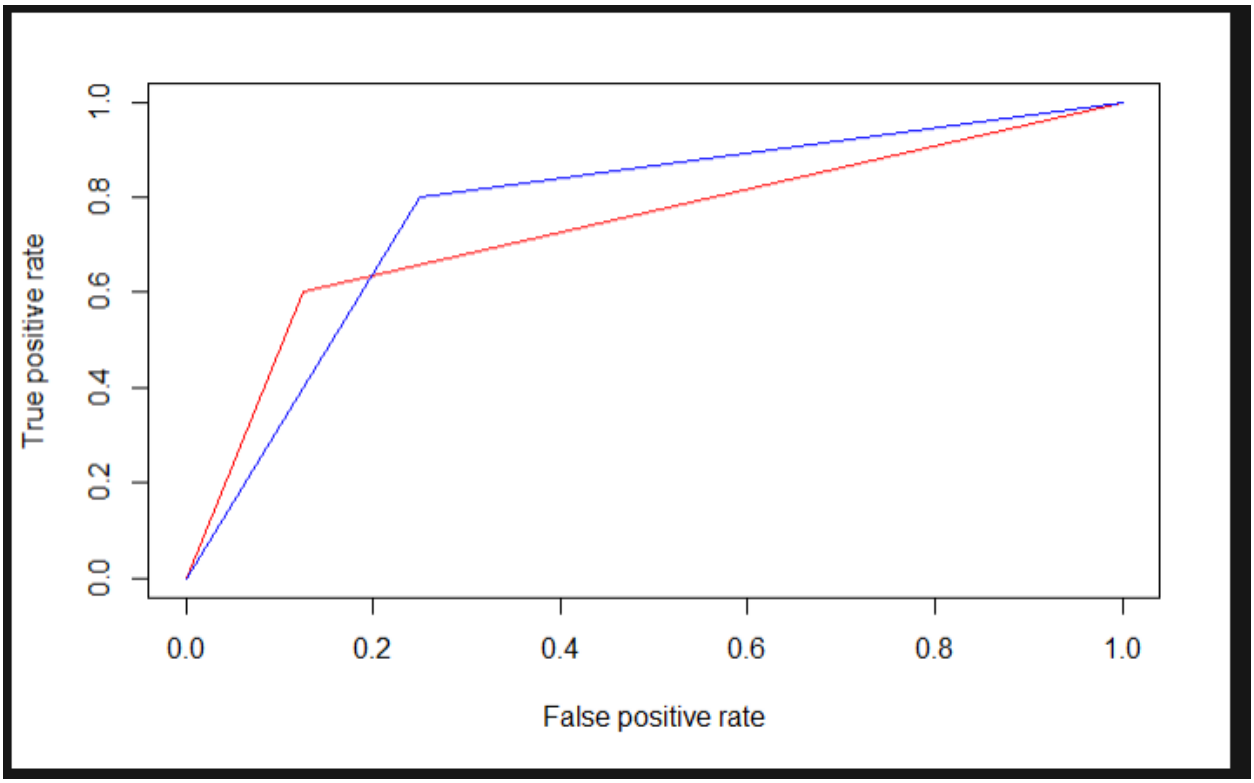
1) The ratio of emails to an individual from a person of interest to all emails addressed to that individual
2) The ratio of emails from an individual to a person of interest to all emails addressed from that individual.
3) The ratio of email receipts shared with a person of interest to all emails addressed to that individual.

Oftentimes, illegal payments are made through bonuses, since salary requires due process such as a contract and a more obvious paper trail, but for bonuses, stringent documentation usually isn't required (*Enron's Many Strands: Executive Compensation, New York Times, 2002)*. So, we created two more features from the financial data to capture this unbalanced pay information:

1) Bonus w.r.t to Salary
2) Bonus w.r.t to Total Payments

Results after Feature Engineering

Model Name	Best Tune	Training Accuracy	Training Precision	Training Recall	Test Accuracy	Test Precision	Test Recall	Test F1	Test AUC
Best Algorithm In 3B - RPart	cp = 0.01	0.9438	0.733	0.916	0.8649	0.5	0.6	0.545	0.775
Best Algorithm in 3C - Random Forest with Ranger	Mtry = 15, Splitrule = Gini, Min.node.size = 1	1	1	1	0.9189	0.6	0.75	0.67	0.79



Comparing the ROC for the best algorithms in 3B and 3C. Colour code: Rpart-blue, RF with Ranger - red.

Testing accuracy for rpart has increased after feature engineering. For random forest, unfortunately, overall results have not increased. This could be due to already having a large number of features relative to a fairly limited sample size. The engineered features could have shown more importance if other features were dropped.

Conclusion

While our overall accuracy of our models was satisfactory, what matters more in this context is correctly identifying persons of interest - the minority class. We also had limited data points to work with - which is generally the case for other real-world problems of this nature. We would need a larger dataset to check the robustness of these models.

In terms of future work, better ways to handle the imbalanced data could be considered with counterfactuals to add robustness. Problems such as these depend on a variety of phenomena that can't be always captured from email and financial data. We might need to consider organizational behaviour and business values such as stance towards varying economic factors and global indicators.

References

1. https://www.technologyreview.com/s/515801/the-immortal-life-of-the-enron-e-mails/
2. https://www.kaggle.com/wcukierski/enron-email-dataset/home
3. https://www.wiley.com/en-us/Creative+Accounting%2C+Fraud+and+International+Accounting+Scandals-p-9780470057650
4. https://www.nytimes.com/2002/03/01/business/enron-s-many-strands-executive-compensation-enron-paid-huge-bonuses-01-experts.html