Identifying Fraud from Enron Emails and Financial Data

By Arpit Kanodia

INTRODUCTION

In 2000, Enron was one of the largest company with total revenue of $111 billion. Fortune named "America's Most Innovative Company" for six consecutive years.

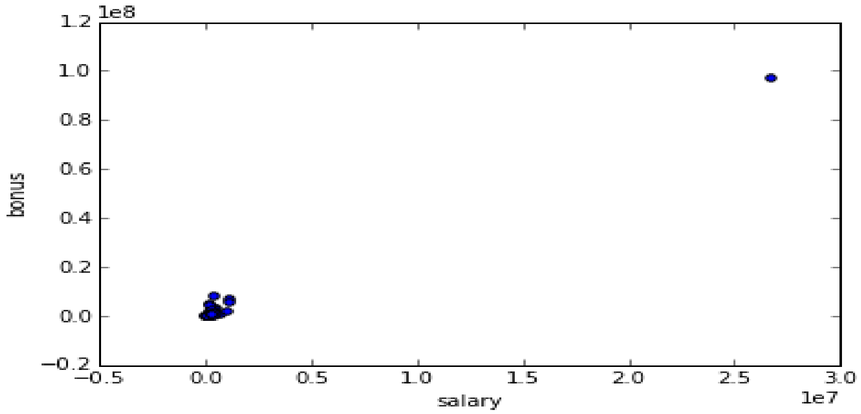
The Enron Scandal revealed in October 2001, that led the company to bankruptcy.

Using scikit-learn and many machine learning methodologies, I created a People of Interest(POI) identifier to predict the culpable persons, using features from financial data, email data and labeled data.

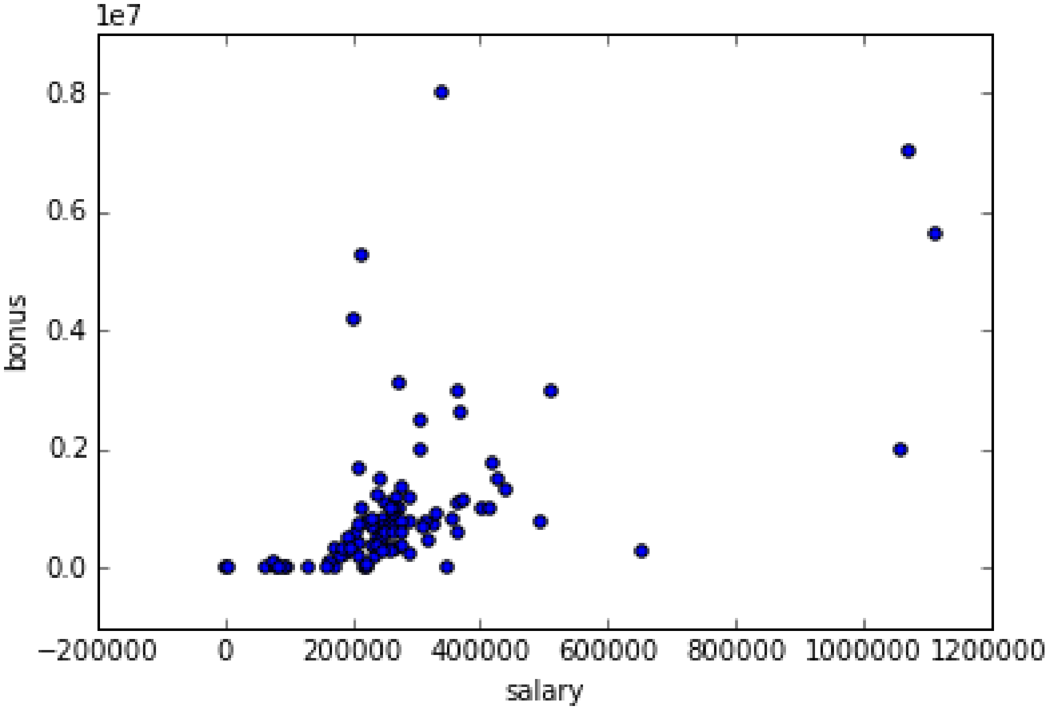
Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those?

The goal of the project is to develop a predictive and analytic model by choosing a combination of features of Former Employees and choose an algorithm that able to predict that a person should be considered as POI or not. The model may provide an application to identify potential suspects for further investigation, then finding proofs against them and ultimately filing charges on them.

The dataset contains 146 records with 14 financial features, 6 email features and 1labeled feature that is POI. For further analysis I created a CSV file and scatter plot.



Clearly TOTAL is a outlier, and need to be removed.



By further analysis of CSV file, I able to find 2 more outliers “THE TRAVEL AGENCY IN THE PARK” and “LOCKHART EUGENE E”

So, the three outliers are removed because of this reasons.

TOTAL : It’s the most extreme outlier with high numerical value, and it is like a lone point in the scatter plot.

THE TRAVEL AGENCY IN THE PARK : This record doesn’t represent the person.

LOCKHART EUGENE E : All the datasets for this data is filled with ‘NAN’.

After removing this outliers dataset contains 143 records.

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 doesn't come ready-made in the dataset--explain what feature you tried to make, and the rationale behind it. If you used an algorithm like a decision tree, please also give the feature importances of the features that you use.

In order to choose the best feature I used the SelectKBest module from scikit-learn for choosing top 10 influential features. (http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html)

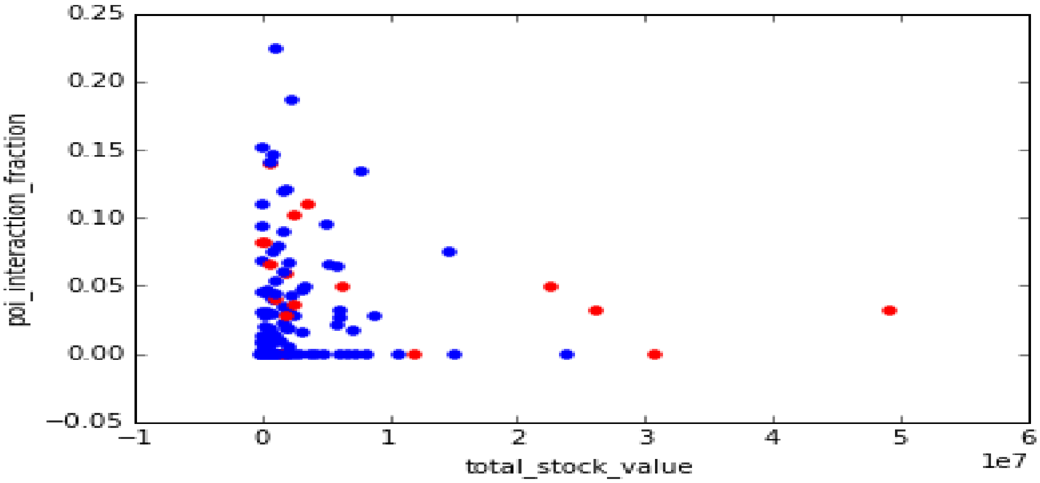
The list of features by there KBest Score.

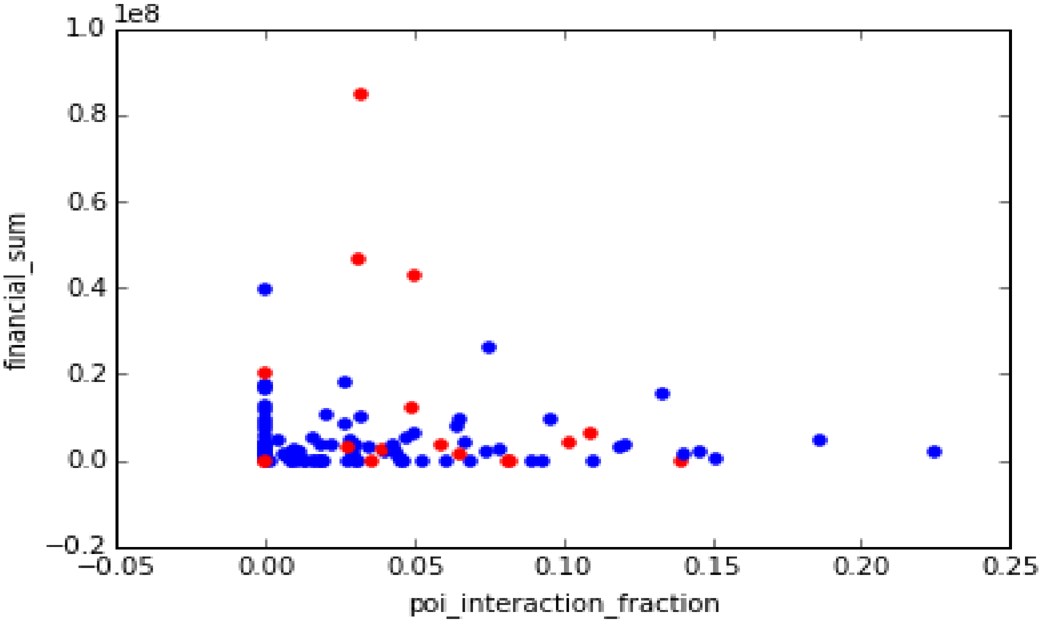
```
[('exercised_stock_options', 24.815079733218194),
 ('total_stock_value', 24.182898678566879),
('bonus', 20.792252047181535),
 ('salary', 18.289684043404513),
('deferred_income', 11.458476579280369),
('long_term_incentive', 9.9221860131898225),
('restricted_stock', 9.2128106219771002),
 ('total_payments', 8.7727777300916792),
 ('shared_receipt_with_poi', 8.589420731682381),
 ('loan_advances', 7.1840556582887247),
('expenses', 6.0941733106389453),
('from_poi_to_this_person', 5.2434497133749582),
('other', 4.1874775069953749),
('from_this_person_to_poi', 2.3826121082276739),
('director_fees', 2.1263278020077054),
('to_messages', 1.6463411294420076),
('deferral_payments', 0.22461127473600989),
('from_messages', 0.16970094762175533),
('restricted_stock_deferred', 0.065499652909942141)]
```

The top 10 most influential features are:-

10 most influential features: ['salary', 'total_payments', 'loan_advances', 'bonus', 'total_stock_value', 'shared_receipt_with_poi', 'exercised_stock_options', 'deferred_income', 'restricted_stock', 'long_term_incentive']

The K best approach is best in automated univariate feature selection, but it lack the email features. For this I created a feature named poi_interaction_fraction which is fraction of total number of emails to and from a POI to the total number of emails sent and received. I also created financial_sum, which is sum of 'total_stock_value', 'exercised_stock_options' and 'salary'. This feature created to simplify and to analyze how much wealth an individual have.





By adding both feature is final analysis, the total number of features selected are 12.

Also before the different machine learning algorithm classifiers, I scaled all features according to max and min.

What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms?

I tried many algorithms KNeighbours, Decision Tree, Gaussian NB etc. I also tried logistic regression (http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

I thought to use this because the prediction outcomes are binary based, i.e. POI or Non-POI.

Algorithm Performance with 12 features

	Accuracy	Precision	Recall
LogisticRegression	0.85187	0.40036	0.22300
KNeighbors	0.86460	0.47237	0.13250
DecisionTree	0.81127	0.28371	0.27250

Clearly, the Accuracy and Precision in KNeighbors are slightly better than LogisticRegression, but there is huge difference in between Recall.

Algorithm Performance with 6 features

	Accuracy	Precision	Recall
LogisticRegression	0.83014	0.30714	0.15050
KNeighbors	0.86429	0.56098	0.23000
DecisionTree	0.78100	0.25595	0.27950

Now, after choosing only 4 most influential features with 2 custom features the KNeighbors is working better, giving better precision and recall. So, I ultimately chooses KNeighbors.

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).

I tuned the parameters by hit and trial method and further by examining different parameters instead of using more complex methods in GridSearchCV. I tried to play with many parameters, I found k refers to the number of surrounding nearest neighbors to look at when voting on majority class. I found best accuracy and precision at k=5.

Further I found the KNeighbours and logistic regression give best result when chooses 12 features (2 custom features).

What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis?

Validation is performed to ensure that a machine learning algorithm generalize the trends well. This reduces the problem of over fitting the model.

I used cross validation from sklearn to use train test split to do the splitting of the data into training set and test set with a test size of 0.3.The num_iters I used is 1100.

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.

Algorithm Performance

	Accuracy	Precision	Recall
LogisticRegression with 12 Features	0.85187	0.40036	0.22300
KNeighbors with 6 Features	0.86429	0.56098	0.23000

The main evaluation metric I used Precision and Recall (but also keeping in mind about decent Accuracy).

Precision depict the actual ratio of true positive records to the records that are actually POI. While, Recall captures the true positive to the records that are flagged as Precision. Both algorithms gave almost similar results. But in my sense Recall is primary metrics for describing the result in this case, a high recall is needed to ensure that truly culpable individual were flagged as POI and investigated thoroughly.