Enron Fraud Person of Interest Identifier

Introduction

Before bankruptcy, Enron was one of the world’s major company in energy trading and was named as one of ”America’s Most Innovative Company”. The investigation after the company’s collapse made a large dataset of company’s email public, which is known as the Enron Corpus.

In this project, I will use the email and financial data of 145 executives to identify the person of interest (POI) in the Enron fraud. A POI is referred to as someone involved in the Enron case, was indicted for fraud, settled with the government, or testified in exchange for Immunity.

Outlier Removal

From the scatter plot of salary and bonus, we can see there is a point in the data set which has much larger value. This point turns out to be the “TOTAL”. It is considered to be an outlier since it is not a real person. This data point is removed manually.

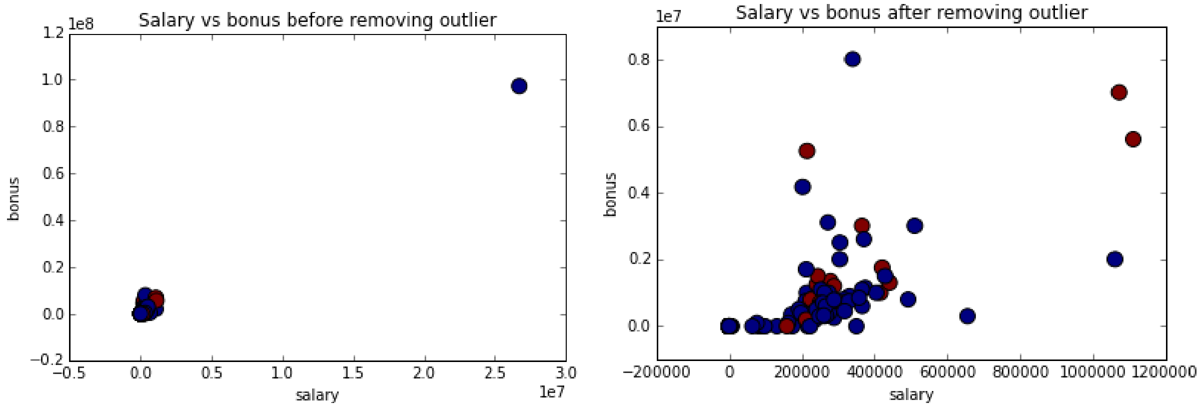

Figure 1,2 Salary and bonus before and after removing outlier

After removing the data, there are still several data points that have much higher value than the rest. Those data represent real people in the Enron case. Some of them are even POI. So they will not be considered outliers.

Feature Scaling

Generally, there are 2 kinds of features. Financial data (salary, bonus, stock value, etc.) and email data (total number of to and from email and number of to and from emails related to POI). These features will be rescaled or recombined to create new features so that they are more effective to distinguish POIs from non-POIs.

The financial data is largely skewed. People like Ken Lay and Jeff Skilling make much more than others. It might create a problem for Naive Bayes algorithm, because the underlying assumption for the algorithm is the distribution should be Gaussian. To resolve this problem the original financial features are taken square root to transform the distribution towards normal distribution.

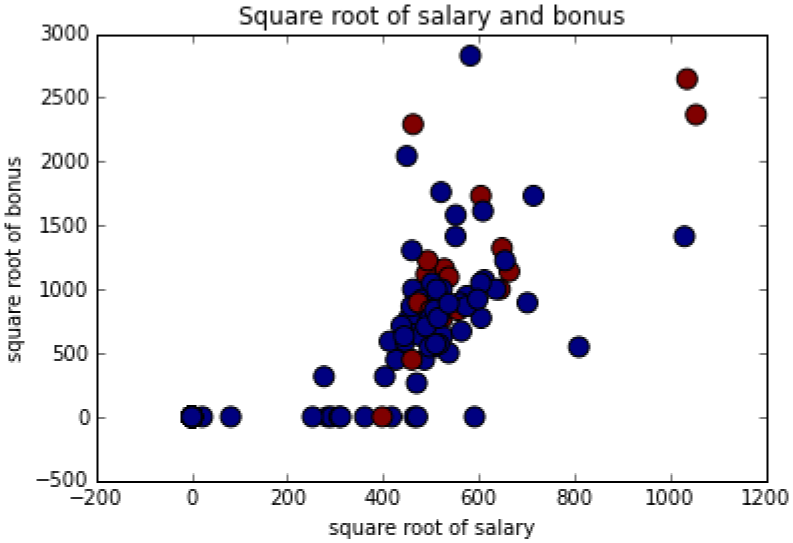

Figure 3 Square root of salary and bonus

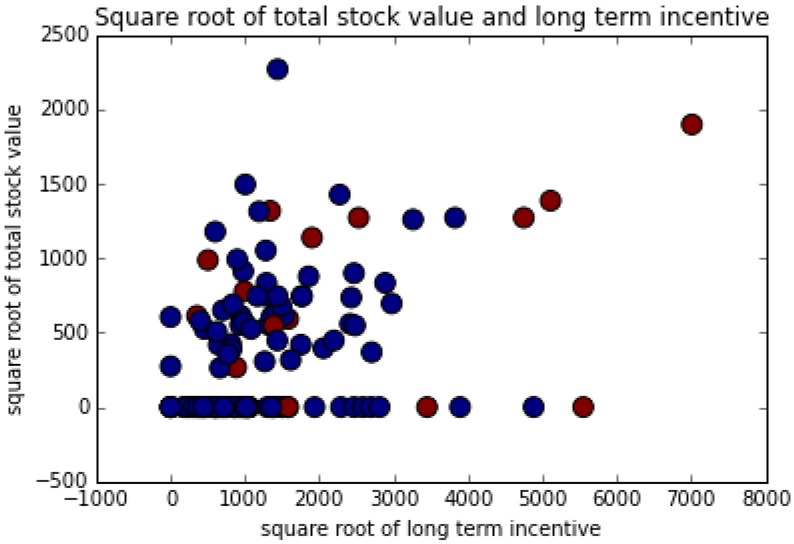

Figure 4 Square root of total stock value and long term incentive

Figure 3 and 4 are square root of financial features. The transformed features looks more normally distributed.

Constructing New Features

We are not only interested in the absolute value of one's income but also the composition of their income. The assumption is that POI and non-POI may have different composition of their income. One way is to look at the ratio between their salary and bonus. POI may get more bonuses compared to their salary.

The other kinds of features are emails data. Instead of looking at the total number of email from or to POI, we are more interested in the portion of email that involves POI. The assumption is that is one involved in fraud are more likely to communicate more frequently with POI.The email data do not contain all the person in the data set. If there is a missing value in the total number of email or the email involves POI. The ratio will be set to 0.

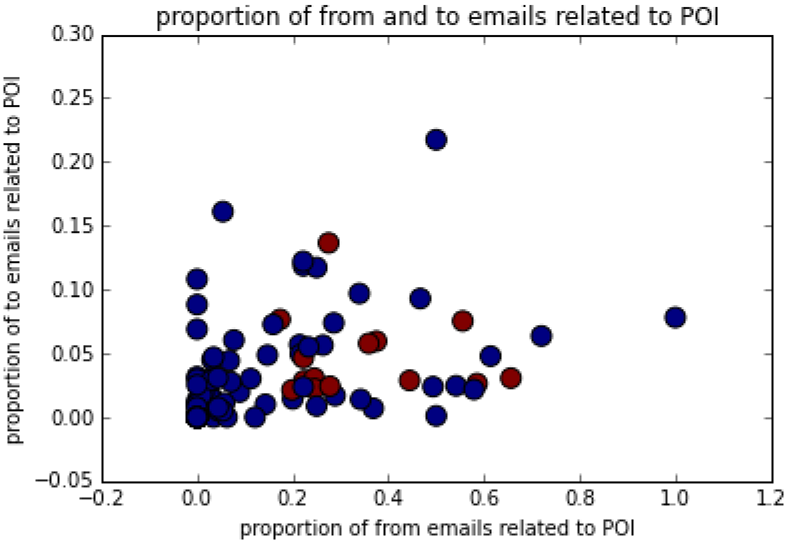

Figure 5 The portion of from emails that involves POI

The list of features used to identifying POIs are:

1. square root of salary
2. square root of bonus
3. square root of total stock value
4. square root of long term incentive
5. The ratio of bonus and salary
6. The portion of from emails that involves POI
7. The portion of to emails that involves POI

Feature Reduction

The fact that the features are correlated makes it beneficial to do feature reduction. For an example, salary and bonus are positively correlated. People who are at higher position are more like to make more salary and bonus. Similarly, people who are closely connected to POI might have higher portion of emails related to POI in both the from and to emails.

We can reduce the number of features using Principal Component Analysis (PCA) which preserves the largest variations. Reduce number of feature might cause the result to deteriorate, the exact number of how many feature is also dependent on the algorithms. It would be decided by trial and error.

Algorithm Selection

Three algorithm are compared with their performance each represent a different kind of algorithm: Naive Bayes (a parametric algorithm), decision tree (a non-parametric algorithm), and random forest (an ensemble method).

Naive Bayes is a parametric method that assumes the distributions of the features are normal. The assumption is that features of POI and non-POI will have different distribution (mean and covariance). The algorithm has that an advantage when the features are in high dimensions. However, the underlying assumption for Naive Bayes is that the features are normally distributed. So the financial features have been rescaled to transfer toward normal distribution.

Decision Trees is a nonparametric algorithm. Unlike Naive Bayes, it does not need the data to be in any particular distribution. So it does not need feature scaling. It can also find irregular decision boundaries. To optimize its performance, a set of tuning parameters need to be chosen such as the minimal sample leaf and sample split.

Random Forests are an ensemble learning method for classification. It searches over a random subset of decision trees. Compared with Decision Trees, Random forests are less likely to overfit the training set.

GridCV was used to automatic search for the best parameter in Decision tree and random forest.

Validation

The validation of the algorithm performance is conducted use the tester function provided (without changing). The function uses cross validation with 1000 folds.

Precision: precision is defined as the number true positive divided by the number of person labels as positive. A higher precision value means a person flag out as a POI is more likely to be a true POI. Recall: recall is defined as the number of true positive divided by the total number of positive. A higher recall value mean if a person is a POI, the algorithm is more likely to flag this person out.

The final results of the algorithm are shown in the table below.

Algorithm	Accuracy	Precision	Recall	# of features (by PCA)	Parameters
Naive Bayes	0.807	0.360	0.332	7	NA
Decision Tree	0.794	0.346	0.381	4	min_samples_leaf=1, min_samples_split=2 *
Random Forest	0.843	0.465	0.151	3	min_samples_leaf=5, min_samples_split=3, n_estimators=5

* Note: the metric for the “best” is different in GridSearch from the metrics in the “tester”. In the tester, we are mainly interested in the precision and recall. The best parameter given by the GridSearch is min_samples_leaf=2, min_samples_split=4. However, this set of parameter does not give the best performance in term of precision and recall.

Discussion

Both Naive Bayes and Decision Trees classifier has achieved precision and recall rate higher than 0.3. Random forest classifier although have the highest accuracy and precision. It does not have good recall, which mean the person flag out by the algorithm as POI are more likely to be the true POI. However, the algorithm see a true POI it is less likely to flag this person out.

There is usually a trade off between precision and recall. An algorithm strong at one metric may be weak at the other metric. So when it comes to decide which algorithm is better, it come to how do we define risk. Is it more risky to flag out as a POI who is actually not or is it more risky to miss a ture POI? That is in the eyes of the beholder.

In this particular problem, Naive Bayes and Decision Trees have more balanced overall performance. One thing about the decision trees need to be mention, it might have a tendency in overfitting, since the number of min sample leaf and min sample split is low. Because the size of the data set is limited (only 18 POIs), although the tester use cross validation, we cannot know for sure whether the data has been overfitted.

Conclusion

In this project, I used machine learning techniques to identify POI in the Enron fraud. Features from both financial and email data are chosen. The original features has been rescaled. New features have been constructed. Feature reduction and algorithm parameter tuning has also been performed. Three algorithm Naive Bayes, Decision Tree and Random Forest was compared. Naive Bayes and Decision Tree have a more balanced overall performance with both precision and recall higher than 0.3.

One of the challenges in this project is that there are no signatures feature that can distinguish POI from non-POI. The are much overlapping “area” between the two classes. The other challenge is small and unbalanced data set, there are only 18 POI in the total 146 people.