Identify fraud from Enron Email
Introduction to Machine Learning Assignment

Vijay Pazheparampil (05-May-2018)

Project Overview

In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, a significant amount of typically confidential information entered into the public record, including tens of thousands of emails and detailed financial data for top executives. In this project, I will play detective, and put my new skills to use by building a person of interest identifier based on financial and email data made public as a result of the Enron scandal.

The objective of my detective task is to build a Person of Interest (PoI) Identifier with the help of email and financial data provided based out of 146 Enron employees. This data includes a hand-generated list of persons of interest in the fraud case , which means individuals who were indicted, reached a settlement or plea deal with the government, or testified in exchange for prosecution immunity.

The steps followed for preparing the PoI Identifier

1) Clean up of Data/Outliers handling
2) Understanding definition of key features and Engineering new features
3) Comparing machine learning algorithms
4) Tuning parameters of the algorithm
5) Validation of analysis
6) Evaluation of algorithms performance

Each of the steps in detailed below:

1. Clean up of Data/Outliers handling

After going through the dataset, I also found that there two rows (THE TRAVEL AGENCY IN THE PARK and LOCKHART, EUGENE E) which seemed irrelvant. The first one looked not like a person and the other one did not have financial details provided.

I keep only employees who have salary details given Out of the 146 employees, i found that there were only 95 employees whose salary is provided. So I remove them from my dataset before I build the model.

Additonally while preparing the scatter plot I found that there is one outlier which was way different from the salary and bonus figures. I found that this outlier was the sum total of the financial figures, so I removed it.

To summarise here is data clean up steps done:

1) Removal of 2 irrelevant data.
2) Removal of 49 employees whose salary details is not provided
3) Removal of TOTAL , which indicates the sum of financial figures.

Finally after the clean up I ended up with **94** datasets which will be used to build my model.

2. Understanding definition of key features and Engineering new features

Available features from the dataset (financial and email data):

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

Based on the definition of the features available in Page 5 of enron61702insiderpay.pdf, the key features I choose:

Existing Financial features : `['salary', 'deferral_payments', 'total_payments', 'exercised_stock_options', 'bonus', 'restricted_stock', 'restricted_stock_deferred', 'total_stock_value', 'expenses', 'loan_advances', 'other', 'director_fees', 'deferred_income', 'long_term_incentive']`

Existing Email features : `['to_messages', 'shared_receipt_with_poi','from_messages', 'from_this_person_to_poi', 'email_address', 'from_poi_to_this_person']`

New Features Defintions I have Introduced:

1) 'Total Stock Value' which is the sum of 'Exercised Stock Options', 'Restricted Stock' and 'Restricted Deferred Stock'
2) 'to_poi_ratio' is the ratio of e-mails sent to PoI
3) 'from_poi_ratio' is the ratio of e-mails received from PoI

In order to find the most effective features for classification, I used SelectKBest to score the features. I got the 10 best features that influences my result:

Feature Engineering Results:

Feature	Score
bonus	9.73
exercised_stock_options	8.43
total_stock_value	8.34
salary	6.94
deferred_income	5.89
total_payments	5.04
loan_advances	4.64
restricted_stock	4.42

long_term_incentive	3.87
Other	1.78

Interestingly, there are no email related features that made into the list. It was also important to note that the newly introduced feature 'total_stock_value' has made it to the list at position 3. It is important to realize that SelectKBest gives us a better idea of the data through univariate feature selection, it doesn't necessarily optimize it, meaning the inclusion of e-mail features into our model may give us more accurate results.

3. Comparing machine learning algorithms

I have used 3 algorithms; Naive Bayes Algorithm , Decision Tree Algorithm and Logistic Regression Alogorithm. The accuracy I got from the models were 0.64, 0.73 and 0.79 respectively.

Algorithm	Accuracy	Precision	Recall
Naive Bayes	0.64	0.16	0.23
Decision Tree (min_samples_split = 6,criterion = entropy)	**0.73**	**0.27**	**0.25**
Logistic Regression (classifier__tol = 1, classifier__C = 0.1)	0.79	0.24	0.09

4.Tuning parameters of the algorithm

Tuning the parameters of an algorithm refers to find the parameters which the highest Accuarcy, Precission and Recall for an algorithm. This might be so effective but it might lead to overfit and get a bad results of the learning process.

Thanks to this wonderful tool , GridSearchCV, I used it to get the best parameter for each algorithm.

Manual tuning included determining which parameters to add to each algorithm and adding/removing features. GridSearchCV provided a convenient way to perform linear combinations for all of the different parameters and report the best result (which we can get using clf.best_estimator_ and clf.best_params_).

Decision Tree parameter : min_samples_split = 6, criterion = entropy
Logistic Regression : classifier__tol = 1, classifier__C=0.1

5. Validation of analysis

Model validation is referred to as the process where a trained model is evaluated with a testing data set. Model validation is carried out after model training. Together with model training, model validation aims to find an optimal model with the best performance.

A classic mistake is also overfitting, it happend when the model performed well in training but not in the test set. In order to avoid this overfitting, I have created a function called evaluateClf which I calculate the mean of accuarcy, precision and recall of 20 different training data.

Since the accuracy of the Logistic Regression is higher compared to the other algorithms I initially throught of considering choosing it. However this is a mistake as the dataset label is quite imbalanced. There are only 18 PoI for 143 records. So I use precision and recall more important here, so I go with Decision Tree Algorithm.

6. Evaluation of algorithm performance

While my decison tree algorithm gave me slightly better results than tester.py.

Algorithm	Accuracy	Precision	Recall
Tester.py	0.68	0.22	0.23
My Decision Tree Algorithm	**0.73**	**0.27**	**0.25**

I have considered precision and recall the most importants parameters.The precision can be interpreted as the likelihood that a person who is identified as a POI is actually a true POI. Here in my PoI identifier it would mean 73% of positive PoI would mean a false alarm. I believe such a low precision is due to the dataset being quite imbalanced. Recall measures how likely it is that identifier will flag a POI in the test set. 25% of time it would the PoI; 75% of time it wouldnt.

One way to improve the identifier would be to use the email data more. We should have featurised more from the email by going into what was written inside those texts. Anyway, this is something I leave it for my future work as I learn and understand machine learning techniques more.

References :

1) http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
2) https://en.wikipedia.org/wiki/Precision_and_recall