Udacity Data Analyst Nanodegree
Project 5: Machine Learning- Identifying Fraud from Enron Email
Rahel Ghebrekidan
August 18, 2016

Introduction

Enron is an American corporation based on Houston Texas. It was an energy, commodity and services company. Enron was among the world's top energy, communication and pulp and paper companies. During 2000, its revenue was claimed to be $111 billion. Suddenly, in late 2001, Enron filed bankruptcy which was considered as a planned accounting fraud. Financial information and emails of Enron is open to public.

The aim of this project is to detect employees who have committed fraud crime based on the financial and emails in the dataset given using machine learning. I will try different classifiers and choose an algorithm which gives best accuracy, precision and recall for our prediction.

Overview of the Dataset

In the dataset we have 146 data point and 21 features which are 14 financial feature and 6 email features and 1 label. The label indicates whether person of interest or not. There are 18 persons of interest and 126 non persons of interest in the dataset. I have removed some records which I considered as outliers. 'Total' which is a column total of all the financial features is removed from the list as it is listed with the employees and does not much with the list. In addition to that, a data point with the name 'The Travel Agency In the Park' is removed because it is not name of employee. I want to keep only names of employees as observations(rows). I have noticed that one of the employee with the name 'Lochart Eugene E' does not have any records so I have removed him from my analysis and prediction. Moreover, I have not included in my feature lists the 'email address' feature because it does not have any significance.

The features have many missing values especially the missing values for 'Defferal payments', 'Loan advanced', 'Restricted stock deferred', 'Deferred Income', 'Long term incentive', and 'Director fees' is more than 50%. I have not replaced the NaN values, I have decided to keep them as they are for this project.

Feature Selection

I have created new features to get better predictors from the email feature. Instead of using the number of email sent to poi (from_thisi_person_to_poi'), and emails received from poi('from_poi_to_this_person'), it is better to use the proportion of emails that a person received or sent compared to all he messages received or sent. 'fraction_to_ poi' and 'fraction_from_poi' are created.

To select features, I have used SelectKbest function, which selects features according to k highest score. In order to help me to select the best list of features, I tried 4 different sets of features by changing the k value. The features selected for each k value and their scores are:-

K = 10

Salary(18.29)
Total_payments: (8.78)
Bonus(20.79)
Total_stock_value (24.18)
Shared_receipt_with_poi (8.60)
Fraction_to_poi(16.60)
Exercised_stock_options (24.82)
Deferred_income(11.49)
Restricted_stock(9.21)
Long_term_incentive: (9.92)

K = 6

Salary(18.29)
Bonus(20.79)
Total_stock_value (24.18)
Fraction_to_poi(16.60)
Exercised_stock_options (24.82)
Deferred_income(11.49)

K=4

Salary(18.29)
Bonus(20.79)
Total_stock_value (24.18)
Exercised_stock_options(24.82)

K=2

Total_stock_value (24.18)
Exercised_stock_options (24.82)

Before deciding how many features to select, I have checked the evaluation metrics of different number of features (k- values) using Decision tree. The results are as follows:

K-Value	Accuracy	Precision	Recall
2	77%	22%	20%
4	79%	32%	33%
6	79%	26%	28%
10	82%	32%	29%

First I have taken ten features but I was not getting good recall values even after tuning the algorithms manual and Grid search. Finally, I decided to go with the best four features where precision and recall are higher (when k= 4). The four best features are ‘salary', 'bonus', 'total_stock_value', and 'exercised_stock_options'.

Classifiers Used

I have tested four algorithms; Decision Tree, Random Forest, KNeighbors and Adaboost. Evaluation Metrics for each algorithm is as below:

Classifier	Accuracy	Precision	Recall
Decision Tree	81%	29%	30%
Random Forest	86%	49%	23%
KNeighbors	86%	21%	3%
AdaBoastClassifier	85%	54%	21%

The result is better when using KNeighbors.

Tuning Algorithm Parameters

Tuning an algorithm is a process of selecting the best set of parameters to optimize a result drawn from an algorithm. I have tuned three algorithms (RandomForest, AdaBoost, and KNeighbors) using Grid search. Grid search is an exhaustive searching through a manually specified subset of parameters. The best evaluation metrics is achieved with AdaBoost Classifier.

Parameters	Evaluation Metrics

Max_depth = 3
Min_sample_split= 2
N-estimators = 100
Criterian= ‘gini’
	Accuracy = 84%
	Precision = 46
	Recall = 33%

Validation

Validation is a process of assessing or testing the performance of a classifier on independent dataset. It is a means of checking an algorithm if it is working properly as planned. The process is done first by training classifier using train data and tested on separated test data. This helps to check overfitting and increases the performance of independent dataset. It is always good to use as maximum data as possible both for train and test. When splitting the data into train and test data, there is a tradeoff. In our case the dataset is very small only 143 observations hence I did cross validation using stratified shuffle split.

Evaluation Metrics

Evaluation Metrics are measures that enable us to see if the algorithm (model) we are using to predict is good algorithms(model). In this analysis the evaluation is derived from cross validating using test data which is allocated by the stratified shuffle split. To get the best results, searching the best set of parameters is done by many trials using the Grid Search. I have made many attempts to get better results.

As mentioned earlier, out of the three algorithms I tried I have found AdaBoost to be the best one. The evaluation Metrics found with that are 84% Accuracy, 46% Precision and 33% Recall. These results indicate that the model's prediction is 84% accurate when classified persons as persons of interest(poi) or non poi. The precession also shows that based on our model, 46% of persons of interest are correctly prediction as d persons of interest. That is if there are 100 persons of interest predicted, using the AdaBoost algorithm, 46 of them are correctly predicted as persons of interest. In addition, the recall result, 33%, shows me that using the algorithm, on average 33% of all persons of interest are labeled correctly. In other words, if there are 100 persons of interest, we can correctly identify the 33 persons of interest. Out of the three evaluation metrics, for this case recall will give better conclusion on identifying the criminals. Hence, based on the financial and email information provided, the model identified 33% of the criminals correctly.

Reference

1. Scikit learn. Machine Learning with Python: Retrieved from http://scikit-learn.org/stable/
2. Raschka, S.(2014). About feature Scaling and Normalization: Retrieved from http://sebastianraschka.com/Articles/2014_about_feature_scaling.html
3. Wikipedia. Enron: Retrieved from https://en.wikipedia.org/wiki/Enron