Enron Fraud Detection Using Machine Learning

Xin Xiao

Project Overview

Enron was one of the largest companies in US. The company, however, collapsed into bankruptcy due to widespread corporate fraud by 2002. There was a significant amount of email and financial information for top executives of Enron entered into the public record. In this project, I will play a role of detective and implement machine learning algorithms to build a person of interest (POI) identifier to detect Enron Employees who may have committed fraud based on the public Enron financial and email dataset.

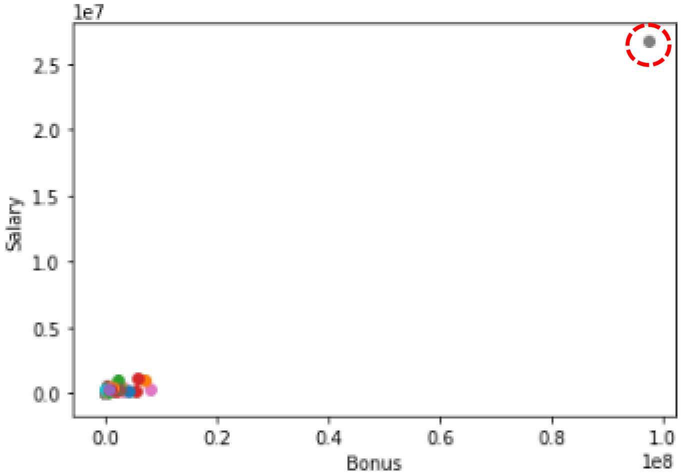
Questions

1. **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?**

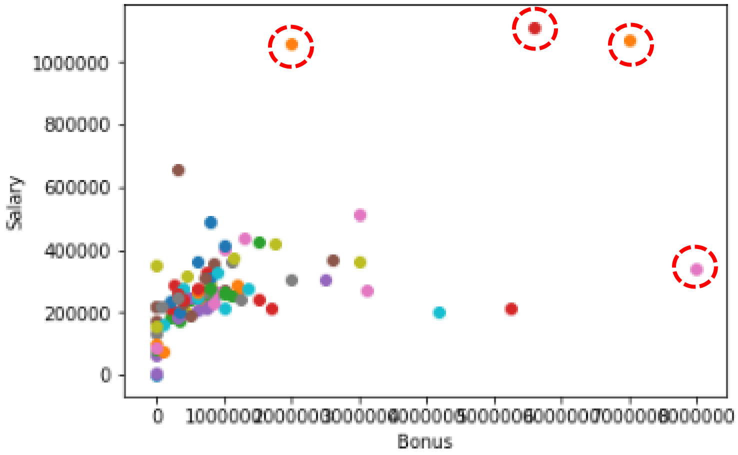
   The goal of this project is to identify POI who may get involved in the Enron fraud scandal. I can use machine learning to build a powerful classifier based on email and financial data to help identify whether a person in the database is a POI or not.

   In this dataset, there are 146 persons in total and each one has a record of 21 features. The actual number of POI is 18 which only takes a percentage of 12.3% in the entire dataset. Thus, this dataset is imbalanced which is important to keep in mind.

   In a scatter plot of ‘bonus’ vs ‘salary’, we can see one outlier clearly in the far upper right corner of the plot. I found that point is ‘TOTAL’, the accumulative value. I removed it from the dataset.



After removing the outlier ‘TOTAL’, I replot the scatter plot and find that there are 4 outliers with substantially high values compared to other data points.



I also find there is one entry with the name ‘THE TRAVEL AGENCY IN THE PARK’, which is not a name. This datapoint is also removed from the dataset.

2. **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 does not come ready-made in the dataset -- explain what feature you tried to make, and the rationale behind it. (You do not necessarily have to use it in the final analysis, only engineer and test it.) In your feature selection step, if you used an algorithm like a decision tree, please also give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values.**

I created two features, namely, ‘‘from_poi_to_this_person_ratio’ and ‘from_this_person_to_poi_ratio’, which characterize the percentages of emails for each person receive and send to POI. The higher percentages of these two features, the more likely they can potentially be POI. The percentage makes more sense compared to the absolute email numbers. In addition, I discard three features (‘total_payments’, ‘total_stock_value’ and ‘email_address’) since they are less relevant.

I used SelectKBest for feature selection. To investigate the importance of each feature, I compare the scores of all the features and select the top 3 features with highest scores as shown in Table I. Therefore, my final feature choices are ‘from_this_person_to_poi_ratio’, ‘expenses’ and ‘salary’.

Table 1   Feature Scores for Top 3 Features

Feature	Score
from_this_person_to_poi_ratio	9.400
expenses	3.975
salary	3.379

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

I tried Naïve Bayes, AdaBoost and K-means clustering on the dataset, and characterized these three algorithms using accuracy, precision, recall and F1 score. Below is the summary table of performance for each algorithm.

Table 2 Performance Comparison of Algorithms

Algorithm	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	0.30	0.13	0.57	0.21
AdaBoost	0.74	0.30	0.43	0.35
K-means clustering	0.86	0.67	0.29	0.40

As shown in table, Naïve Bayes has the lowest accuracy and F1 score although its recall is the highest among three algorithms, which is not promising. AdaBoost and K-means clustering both show pretty high accuracies and F1 scores, showing superior performance to identify POI.

Since both precision and recall need to be at least 0.3, I choose AdaBoost as my final classifier.

4. **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? What parameters did you tune? (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).**

For a machine learning algorithm, there are usually many parameters in the model which controls the overall performance of algorithms, i.e. affecting accuracy, precision and recall etc. Thus, it is of importance to tune these algorithm parameters so that it can provide the best performance.

I tuned the parameters for K-means clustering and AdaBoost using GridSearchCV module and a stratified shuffle split on a thousand folder. For K-means clustering, I tuned two parameters, n_init and tol (n_cluster is always 2). For AdaBoost, three parameters, n_estimators, learning_rate and algorithm, were tuned.

5. **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 the process of training and testing a machine learning algorithm to access its performance. If the validation process is conducted in a wrong manner, the overfitting is likely to occur, that is said, the trained model performs well on the training dataset but significantly worse on the cross-validation and test datasets.

   In addition to overfitting, it is also important to perform random shuffling on the entire dataset and then split the dataset into training and test datasets.

   In this project, I used the tester function provided in the course and GridSearchCV function with StraitifiedShuffleSplit of 1000 folds to validate the analysis and also identify optimal combination of parameters.

6. **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.**

   In this project, I used four evaluation metrics, namely accuracy, precision, recall and F1 score, to characterize the performance of algorithms. Accuracy is a direct measure on the percentage of how many POIs are identified from the dataset. Precision is defined as the ratio of true positives to the entries that are actually POIs. A high precision value means POI identified by the algorithm is highly likely to be correct and a low value means there are many false alarms, that is non-POI is flagged as POI. Recall refers to the ratio of true positive to the entries flagged as POIs, which is used to characterize the sensitivity of algorithm. A high recall value means that the algorithm is able to find more POIs from the dataset while a low value means the algorithm is difficult to identify POIs. F1 score is a weighted average of precision and recall, ranging from 0 to 1. It is widely used to characterize the accuracy of a machine learning algorithm.