Enron Mail Fraud Detection, Udacity Data Analyst Nanodegree

Shreyas Ramnath 7th January 2018

Question 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 was to use financial and email data from [Enron corporation] (https://www.cs.cmu.edu/~./enron/) - publicly made famous by the US Federal Energy Regulatory Commission during its investigation of Enron, which comprised email and financial data of 146 people (records), most of which are senior management of Enron, to come up with a predictive model that could spot an individual as a "Person of Interest (POI). The corpus is widely used for various machine learning problem and although it has been labeled already, the value is the potential application for similar cases in other companies or spam filtering application. The dataset contained 146 records with 1 labeled feature (POI), 14 financial features, 6 email features. Within these records, 18 were labeled as a "Person of Interest" (POI).

However, the dataset contains numerous people missing value for each feature which can be described as table below:

| Feature | Not a Number per feature |

| :--- | :--: |

| Loan advances | 142 |

| Director fees | 129 |

| Restricted stock deferred | 128 |

| Deferred payment | 107 |

| Deferred income | 97 |

| Long term incentive | 80 |

| Bonus | 64 |

| Emails sent also to POI | 60 |

| Emails sent | 60 |

| Emails received | 60 |

| Emails from POI | 60 |

| Emails to POI | 60 |

| Other | 53 |

| Expenses | 51 |

| Salary | 51 |

| Exercised stock option | 44 |

| Restricted stock | 36 |

| Email address | 35 |

| Total payment | 21 |

| Total stock value | 20 |

Enron was among Fortune 500 in U.S in 2000. By 2002, it collapsed due to corporate fraud resulting from Federal Investigation; there were thousands of records (e-mail & financial data). Most notable of which are Jeffery Skilling, Key Lay, and Fastow all have dumped large amounts of stock options, and they are all deemed guilty.

I used `scikit-learn` & various machine learning techniques to predict "Person of Interest", detecting culpable person using both financial & email-data. Through exploratory data analysis and CSV check, I was able 3 records need to be removed:

- `TOTAL`: Through visualizing using scatter-plot matrix. We found `TOTAL` are the extreme outlier since it comprised every financial data in it.

- `THE TRAVEL AGENCY IN THE PARK`: This must be a data-entry error that it didn't represent an individual.

- `LOCKHART EUGENE E`: This record contained only NaN data.

Question 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.

In the cleaned data, I chose 'from_poi_to_this_person' and 'from_this_person_to_poi' features. But suprisingly no strong pattern was seen when I plotted the data. Hence as a workaround so I used fractions for both features of “from/to poi messages” and “total from/to messages” to plot the data. Here I had no idea about which feature to chose according to their importance

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

Here I used Decision Trees to rank the features according to importance. But the selection of features was a repetitive process. All the possible features were pushed into a list called; features_list and I then started eliminating them one by one using score value and my intuition. With these features my recall and precision were too low (less than 0.3) , hence I changed my method and manually picked out features which gave me precision and recall values over 0.3. I could not use accuracy for evaluating the algorithm due the scarcity of POI's in this dataset and the best evaluators available to me were precision and recall. There were only 18 examples of POIs in the dataset. There were 35 people who were POIs in "real life", but for various reasons, half of those are not present in this dataset. Finally I picked the following features:

["frac_from_poi_email", "frac_to_poi_email", "shared_receipt_with_poi"]

Question 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?

Parameters tuning refers to the adjustment of the algorithm when training, in order to improve the fit on the test set. Parameter can influence the outcome of the learning process, the more tuned the parameters, the more biased the algorithm will be to the training data & test harness. The strategy can be effective but it can also lead to more fragile models & over fit the test harness but don't perform well in practice

With every algorithm, I tried to tune as much as I could with only marginal success & unremarkable improvement but come up with significant success with Decision Trees. Manually searching through the documentation, I came up with these following parameters:

class sklearn.tree.**DecisionTreeClassifier**(*criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *class_weight=None*, *presort=False*)

Question 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 comprises set of techniques to make sure our model generalizes with the remaining part of the dataset. A classic mistake, which was briefly mistaken by me, is over-fitting where the model performed well on training set but have substantial lower result on test set. In order to overcome such classic mistake, we can conduct cross-validation (provided by the `evaluate` function in `poi_id.py`).

Question 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

For this assignment, I used `precision` & `recall` as 2 main evaluation metrics. The best performance belongs to decision tree algorithm (`precision: 0.6667` & `recall: 0.6667`) which is also the final model of choice, as decision trees are widely used in text classification; we can actually extend this model for email classification if needed. Precision refer to the ratio of true positive (predicted as POI) to the records that are actually POI while recall described ratio of true positives to people flagged as POI. Essentially speaking, with a precision score of 0.6667, it tells us if this model predicts 100 POIs, there would be 67 people are actually POIs and the rest 33 are innocent. With recall score of 0.6667, this model finds 66.7% of all real POIs in prediction. This model is amazingly perfect for finding bad guys without missing anyone, but with 33.33% probability of wrong. . Due to the nature of the dataset, `accuracy` is not a good measurement as even if non-POI are all flagged, the accuracy score will yield that the model is a success.

References:

- [Introduction to Machine Learning (Udacity)]

- [MITx Analytics Edge]

- [Scikit-Learn Documentation]