Identify Fraud from Enron Mail

Abhishek Babuji

JULY 11, 2017

1 Introduction

The Enron scandal, publicized in October 2001, eventually led to the bankruptcy of the Enron Corporation. The company would build an asset, such as a power plant, and immediately claim the projected profit on its books, even though it hadn't made one dime from it. If the revenue from the power plant were less than the projected amount, instead of taking the loss, the company would then transfer these assets to an off-the-books corporation, where the loss would go unreported.

This is a machine learning project to identify person of interest by trying to uncover patterns in the dataset that contains `126 (non-poi) and 18 (poi)` after removing outliers.

2 Dataset

2.1 Outliers

1. `TOTAL`: This does not refer to a valid datapoint. All observations are persons, and this seems to be nothing but a summary.
2. `THE TRAVEL AGENCY IN THE PARK`: Does not represent a person.
3. `LAY KENNETH L`: He was the CEO and chairman of Enron Corporation for most of its existence and is a central figure in the Enron scandal. The observations in other columns that represent him are anomalies and not outliers.
4. `FREVERT MARK A`: He used to be the Vice - Chairman of Enron; he is an anomaly too.
5. `WHALLEY LAWRENCE G`: He was the the COO and president of Enron Corp. for a short time. Another anomaly.
6. `SKILLING JEFFREY`: Another former CEO. He probably was in the top management position at the time.
7. `LAVORATO JOHN J`: President and CEO, Enron Americas; another extremely high-profile individual who received large bonuses before the company declared bankruptcy.
8. `MCMAHON JEFFREY`: He used to fill in the role of Treasurer at the time.

`TOTAL` and `THE TRAVEL AGENCY IN THE PARK` were removed from the dataset.

2.2 Missingness

Among 144 observations, 19 features, and 1 label named `poi`. There are missing values for many observations. The number and percentage of missing values for each feature is depicted in the table below.

feature	Number of missing values	Percentage of missing values
loan_advances	141	97.916667
director_fees	128	88.888889
restricted_stock_deferred	127	88.194444
deferral_payments	106	73.611111
deferred_income	96	66.666667
long_term_incentive	79	54.861111
bonus	63	43.750000
from_messages	58	40.277778
from_this_person_to_poi	58	40.277778
shared_receipt_with_poi	58	40.277778
to_messages	58	40.277778
other	53	36.805556
expenses	50	34.722222
salary	50	34.722222
exercised_stock_options	43	29.861111
restricted_stock	35	24.305556
email_address	33	22.916667
total_payments	21	14.583333
total_stock_value	19	13.194444
poi	0	0.000000

2.3 New features

1. `ratio_from_poi: from_poi_to_this_person/to_messages`
2. `ratio_to_poi: from_this_person_to_poi/from_messages`

These two features capture the fraction of to and from messages send between a `poi` and the person in focus.

In this project, before fitting classifiers, for personal ease of use; like missing value imputation, outlier detection, the dictionary was converted to a data frame. Classifiers were however fit on dictionaries that were converted from data frames. Machine learning classifiers were tested on a combination of data frames that had extra features and followed different methods for imputing missing values.

DATASETS			
S.NO	Any removed features?	Any added features?	Missing value imputations for other numerical features?
A	All features having more than 40.277% missing	Two newly created features specified above	K-Nearest Neighbors (K=2)
B	All features having more than 40.277% missing	Two newly created features specified above	NaN replaced with 0
C	All features having more than 40.277%	None	K-Nearest Neighbors (K=2)
D	All features having more than 40.277%	None	NaN replaced with 0
E	None	Two newly created features specified above	K-Nearest Neighbors (K=2)
F	None	Two newly created features specified above	NaN replaced with 0
G	None	None	K-Nearest Neighbors (K=2)
H	None	None	NaN replaced with 0

Three algorithms were tuned and fit for all 8 datasets scaled with `MinMaxScaler` using `GridSearchCV`.

3 Parameter Tuning

The ultimate goal of machine learning is to make a machine system that can automatically build models from data without requiring tedious and time consuming human involvement. As you recognize, one of the difficulties is that learning algorithms (eg. decision trees, random forests, clustering techniques, etc.) require you to set parameters before you use the models (or at least to set constraints on those parameters). How you set those parameters can depend on a whole host of factors. That said, your goal, is usually to set those parameters to so optimal values that enable you to complete a learning task in the best way possible. Thus, tuning an algorithm or machine learning technique, can be simply thought of as process which one goes through in which they optimize the parameters that impact the model in order to enable the algorithm to perform the best. The performance of a machine learning classifier can be understood through various evaluation metrics like Accuracy, Precision, Recall, F1 and F2 scores. The definition of what is best, is subjective, and depends on what you want the machine learning classifier to achieve.

To make it more concrete, here is an example. If you take a machine learning algorithm for clustering like KNN, you will note that you, as the programmer, must specify the number of K's in your model (or centroids), that are used. How do you do this? You tune the model. There are many ways that you can do this. One of these can be trying many many different values of K for a model, and looking to understand how the evaluation metrics change as you vary the number of K's in your model.

The most common problems in parameter tuning is 'overtuning'. This will appear to be improving the model's performance on test data but could actually harm generalization if taken too far. One of the indicators that you could be overtuning a classifier is that you will see the classifier achieve a 100% Accuracy on test data set, but work terribly on future data on which the same model will be fit.

4 Result Summary

4.1 GaussianNB

Parameters tuned:

1. SelectKBest (k): From 3 to max_features (varies for every dataset)
2. PCA (n_components): For every k in SelectKBest, n_components = 1 to (k-1)
3. Classifier fit: best_estimator_

S.NO	k, n_components	features	Accuracy	Precision	Recall
A	12, 3	exercised_stock_options, total_stock_value, ratio_to_poi, salary, total_payments, shared_receipt_with_poi, restricted_stock, from_poi_to_this_person, ratio_from_poi, other, from_this_person_to_poi, to_messages	0.84007	0.37461	0.29800
B	9, 5	exercised_stock_option, total_stock_value, salary, ratio_to_poi, restricted_stock, total_payments, shared_receipt_with_poi, expenses, from_poi_to_this_person	0.81940	0.31053	0.29050
C	12, 2	exercised_stock_options, total_stock_value, salary, total_payments, shared_receipt_with_poi, restricted_stock, from_poi_to_this_person, other, from_this_person_to_poi, to_messages, expenses, from_messages	0.84520	0.40254	0.33250
D	12, 5	exercised_stock_options, total_stock_value, salary, restricted_stock, total_payments, shared_receipt_with_poi, expenses, from_poi_to_this_person, other, from_this_person_to_poi, to_messages, from_messages	0.82280	0.33599	0.33700
E	12, 4	exercised_stock_options, total_stock_value, ratio_to_poi, salary, deferred_income, bonus, total_payments, shared_receipt_with_poi, loan_advances, restricted_stock, long_term_incentive, from_poi_to_this_person	0.84387	0.39674	0.32850
F	13, 5	exercised_stock_options, total_stock_value, bonus, salary, ratio_to_poi, deferred_income, long_term_incentive, restricted_stock, total_payments, shared_receipt_with_poi, loan_advances, expenses, from_poi_to_this_person	0.82207	0.32569	0.31250
G	18, 9	exercised_stock_options, total_stock_value, salary, deferred_income, bonus, total_payments, shared_receipt_with_poi, loan_advances, restricted_stock, long_term_incentive, from_poi_to_this_person, other, from_this_person_to_poi, to_messages, director_fees, expenses, deferral_payments, restricted_stock_deferred	0.83453	0.36566	0.32800
H	5, 3	exercised_stock_options, total_stock_value, bonus, salary, deferred_income	0.84540	0.37944	0.25100

4.2 KNeighborsClassifier

Parameters tuned:

1. SelectKBest (k): From 3 to max_features (varies for every dataset)
2. PCA (n_components): For every k in SelectKBest, n_components = 1 to (k-1)
3. KNeighborsClassifier (n_neighbors): For every k and every n_components, n_neighbors is tested from 1 to 10.
4. Classifier fit: best_estimator

S.NO	k, n_components, n_neighbors	features	Accuracy	Precision	Recall
A	7, 6, 1	exercised_stock_options, total_stock_value, ratio_to_poi, salary, total_payments, shared_receipt_with_poi, restricted_stock	0.83360	0.36823	0.34650
B	12, 2, 1	exercised_stock_options total_stock_value, salary, ratio_to_poi, restricted_stock, total_payments, shared_receipt_with_poi, expenses, from_poi_to_this_person, other, ratio_from_poi, from_this_person_to_poi	0.81333	0.27949	0.25350
C	3, 1, 1	exercised_stock_options, total_stock_value, salary	0.81333	0.27949	0.25350
D	4, 1, 3	exercised_stock_options, total_stock_value, salary, restricted_stock	0.81253	0.27766	0.25350
E	6, 1, 1	exercised_stock_options, total_stock_value, ratio_to_poi, salary, deferred_income, bonus	0.80380	0.23644	0.21150
F	5, 2, 1	exercised_stock_options, total_stock_value, bonus, salary, ratio_to_poi	0.81613	0.26691	0.21700
G	14, 10, 1	exercised_stock_options, total_stock_value, salary, deferred_income, bonus, total_payments, shared_receipt_with_poi, loan_advances, restricted_stock, long_term_incentive, from_poi_to_this_person, other, from_this_person_to_poi, to_messages	0.82320	0.28609	0.21800
H	3, 1, 1	exercised_stock_options, total_stock_value, bonus	0.82687	0.33626	0.30650

4.3 LogisticRegression

Parameters tuned:

1. SelectKBest (k): From 3 to max_features (varies for every dataset)
2. PCA (n_components): For every k in SelectKBest, n_components = 1 to (k-1)
3. LogisticRegression (C): For every k and every n_components, the following values of C are tested `[0.0001, 0.001, 0.01, 1, 10, 100]`
4. Classifier fit: best_estimator

S.NO	k, n_components, C	features	Accuracy	Precision	Recall
A	3, 2, 100	exercised_stock_options, total_stock_value, ratio_to_poi	0.87367	0.56318	0.23400
B	8, 7, 100	exercised_stock_options, total_stock_value, salary, ratio_to_poi, restricted_stock, total_payments, shared_receipt_with_poi, expenses	0.86467	0.48171	0.19750
C	12, 10, 100	exercised_stock_options, total_stock_value, salary, total_payments, shared_receipt_with_poi, restricted_stock, from_poi_to_this_person, other, from_this_person_to_poi, to_messages, expenses, from_messages	0.86753	0.50889	0.18600
D	5, 4, 100	exercised_stock_options, total_stock_value, salary, restricted_stock, total_payments	0.87060	0.54766	0.16950
E	3, 2, 100	exercised_stock_options, total_stock_value, ratio_to_poi	0.86700	0.50289	0.21750
F	12, 2, 100	exercised_stock_options, total_stock_value, bonus, salary, ratio_to_poi, deferred_income, long_term_incentive, restricted_stock, total_payments, shared_receipt_with_poi, loan_advances, expenses	0.86973	0.54078	0.15250
G	19, 13, 100	exercised_stock_options, total_stock_value, salary, deferred_income, bonus, total_payments, shared_receipt_with_poi, loan_advances, restricted_stock, long_term_incentive, from_poi_to_this_person, other, from_this_person_to_poi, to_messages, director_fees, expenses, deferral_payments, restricted_stock_deferred, from_messages	0.85013	0.35308	0.14900
H	10, 4, 10	exercised_stock_options, total_stock_value, bonus, salary, deferred_income, long_term_incentive, restricted_stock, total_payments, shared_receipt_with_poi, loan_advances	0.86760	0.51373	0.13100

5 Evaluation Metrics and Validation

There are 3 metrics for evaluation used in this project:

$$\text{Accuracy} = \frac{\text{Number of people that are accurately predicted as POI or accurately predicted as non-POI}}{\text{Number of all people in the dataset}}$$

$$\text{Precision} = \frac{\text{Number of people that are accurately predicted as POI}}{\text{Number of people that are predicted as POI}}$$

$$\text{Recall} = \frac{\text{Number of people that are accurately predicted as POI}}{\text{Number of people are actually POI}}$$

In this dataset,there are `126 (non-poi) and 18 (poi)` after removing outliers. If we predicted all 144 observations as `non-poi`, then that would itself yield an Accuracy of 85.71% due to very less number samples that are `poi`. In layman's terms, its hard to wrongly predict a `non-poi` since it represents 85.71% of the dataset. Recall is the probability of our algorithm to correctly identify `poi`, given that it actually is a `poi`, we want this probability to be as high as possible. In layman's terms; out of all the items that are truly `poi`, how many were correctly classified as `poi`. Or simply, how many `poi` items were 'recalled' from the dataset. Precision in layman's terms can be understood like this; out of all the items identified as `poi`, how many truly belong to the `poi` class.

To avoid overfitting, we split put dataset into training set, and test set. Since our dataset has 144 samples, 126 in class `'non-poi'` and 18 in class `'poi'`. In this unbalanced distribution of the two classes, if you use a normal randomized train-test split, you could end up building models on exceedingly few (or EVEN NONE!) from the `'poi'` class. If you are building a model that is trained on data where there are very few samples in `'poi'` class, how could you expect the model to predict the rarer group effectively? The `StratifiedShuffleSplit` allows for randomization but also makes sure these unbalanced datasets have some of both classes. To avoid that our performance is due to chance, I use 10-fold cross-validation. In other words, for every machine learning algorithm I randomly split the data into training and testing dataset 10 times, and fit algorithm and calculate performance 10 times for all combinations of parameters, and store the average performance; as performance of the algorithm.

6 Impact of adding new features to the dataset

Taking a look at the results of GaussianNB classifier fit to the datasets F and H, where F is a dataset that had no removed features and two newly added features namely and `ratio_to_poi` and `ratio_from_poi`, we see that the classifier had higher recall for this dataset as opposed to dataset H that used all existing features and no newly added features.

6.1 GaussianNB

Parameters tuned:
1. SelectKBest (k): From 3 to max_features (varies for every dataset)
2. PCA (n_components): For every k in SelectKBest, n_components = 1 to (k-1)
3. Classifier fit: best_estimator_

The table below will demonstrate the differences in the evaluation metric scores, for the two datasets F and H.

S.NO	k, n_components	features	Accuracy	Precision	Recall
F	13, 5	exercised_stock_options, total_stock_value, bonus, salary, ratio_to_poi, deferred_income, long_term_incentive, restricted_stock, total_payments, shared_receipt_with_poi, loan_advances, expenses, from_poi_to_this_person	0.82207	0.32569	0.31250
H	5, 3	exercised_stock_options, total_stock_value, bonus, salary, deferred_income	0.84540	0.37944	0.25100

The following tables show the features chosen by SelectKBest and their respective scores. This gives an idea of the impact of adding the new feature to the dataset.

S.NO	k, n_components	features according to SelectKBest	Score
F	13, 5	exercised_stock_options	24.82
		total_stock_value	24.18
		bonus	20.79
		salary	18.29
		ratio_to_poi	16.41
		deferred_income	11.46
		long_term_incentive	9.92
		restricted_stock	9.21
		total_payments	8.77
		shared_receipt_with_poi	8.59
		loan_advances	7.18
		expenses	6.09
		from_poi_to_this_person'	5.24

One of the newly added features `ratio_to_poi` was selected among 13 features for dataset F according to SelectKBest and ranks 5th with a score of 16.41. Let's now have a look at dataset H which only used existing features.

S.NO	k, n_components	features according to SelectKBest	Score
H	5, 3	exercised_stock_options	24.82
		total_stock_value	24.18
		bonus	20.79
		salary	18.29
		deferred_income	11.46

From this table it is evident that `ratio_to_poi` feature that was selected by SelectKBest for dataset F scored more than `deferred_income`, `16.41 > 11.46`

7 Final algorithm

In summary, I would choose `GaussianNB` as the classifier, on dataset C, with features chosen from `best_estimator_`; `[exercised_stock_options, total_stock_value, salary, total_payments, shared_receipt_with_poi, restricted_stock, from_poi_to_this_person, other, from_this_person_to_poi, to_messages, expenses, from_messages]`

8 References

1. Introduction to Machine Learning with Python
2. http://adalee2future.github.io/udacity_data_analyst/: Inspired document formatting
3. Stack Overflow
4. Udacity Forum Mentors: Myles Cannon, thomas_1405649566 and andrew
5. https://stackoverflow.com/questions/22903267/what-is-tuning-in-machine-learning: Parameter Tuning