Machine Learning Project

Enron Fraud Detectors using Enron Emails and Financial Data.

About Enron

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, there was a significant amount of typically confidential information entered into public record, including tens of thousands of emails and detailed financial data for top executives. Most notable of which are Jefferey Skilling, Key Lay, and Fastow all have dumped large amounts of stock options, and they are all deemed guilty.

Goal of the Project

The goal of this project is to use financial and email data from Enron corpus - publicly made by US Federal Energy Regulatory Comission during its investgiation 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 "Person of Interest" here refers to a person who was involved in the fraud. The dataset contained 146 records with 1 labeled feature (POI), 14 financial features, 6 email feature. Within these record, 18 were labeled as a "Person Of Interest" (POI).

The features in the data fall into three major types, namely financial features, email features and POI labels.

- **Financial Features**: salary, deferral_payments, total_payments, loan_advances, bonus, restricted_stock_deferred, deferred_income, total_stock_value, expenses, exercised_stock_options, other, long_term_incentive, restricted_stock, director_fees. (All units are in US dollars)
- **Email Features**: to_messages, email_address, from_poi_to_this_person, from_messages, from_this_person_to_poi, shared_receipt_with_poi. (units are generally number of emails messages; notable exception is ‘email_address’, which is a text string)
- **POI Label**: poi. (boolean, represented as integer)

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

Feature	NaN 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
Excersised stock option	44
Restricted stock	36
Email address	35
Total payment	21
Total stock value	20

Outliers in the data

Through exploratory data analysis which included plotting of scatterplots using `matplotlib`, validating the list of keys in the data dictionary etc, I was able to figure out the following outliers -

- `TOTAL` : Through visualising 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.

Feature Scaling and Selection

Removing Features

By apply some intuition, the following facts could be formulated :

- `email_address` can in no way differentiate between a POI and a non POI.
- Features like `restricted_stock_deferred` , `director_fees`, `loan_advances` contains majority of their values as *NaN*.

Hence, these four features must be removed in order to gain better results.

Creating New Features

Instead of training the algorithm with the features like `from_this_person_to_poi` and `from_messages`, the ratio of number of messages from this person to poi and the total number of messages from this person would be more appropriate. Similar is the case with `from_poi_to_this_person` and `to_messages`.

Main purpose of composing ratio of POI message is we expect POI contact each other more often than non-POI and the relationship could be non-linear. The initial assumption behind these features is: the relationship between POI is much more stronger than between POI and non-POIs

New Features Created - `fraction_from_this_person_to_poi` and `fraction_from_poi_to_this_person`

Features Removed - `from_this_person_to_poi`, `from_poi_to_this_person`, `from_messages`, `to_messages`

Scaling Features

I have used `min_max_scaler` from `preprocessing` module of `scikit-learn` to scale the features. Scaling of features becomes crutial when the units of the features are not same.

Feature Selection

I used `scikit-learn SelectKBest` to select best 10 influential features and used those featuers for all the upcoming algorithm. Unsurprisingly, 8 out of 10 features are related to financial data and only 2 features are related to e-mail.

Selected Features	Score
exercised_stock_options	8.772
total_stock_value	11.458
bonus	24.815
salary	18.289
deferred_income_	9.922
long_term_incentive	16.409
restricted_stock_	20.792
total_payments_	0.225
shared_receipt_with_poi_	9.212
fraction_from_this_person_to_poi_	3.128

Spliting Data into Training and Testing Datasets

In order to split the data into training and testing datasets, I've used `train_test_split` from `cross_validation` module of `scikit-learn`.

```
features_train, features_test, labels_train, labels_test = \
    train_test_split(features, labels, test_size=0.3, random_state=42)
```

The paramenter "test-size" adjusts the size of the testing dataset. In this case, 30% of the dataset is being used for testing.

Trying Different Classifier Algorithms and Evaluating

Algorithms

The classifier I used in order to make the predictions are `GaussianNB`, `Support Vector Classifier`, `Decision Tree Classifier`, `AdaBoost Classifier` and `Random Forest Classifier`.

Evaluation

The evaluation has been done by comparing the recall and precision obtained by using different classifier algorithms. `recall_score`, `precision_score` and `accuracy_score` has been used from `metrics` module of `scikit-learn`.

Parameter Tuning

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 & overfit the test harness but don't perform well in practice

Final Model of Choice

For this assignment, I used precision & recall as 2 main evaluation metrics. The best performance belongs to Decision Tree Classifier **(precision: 0.314& recall: 0.587)** which is also the final model of choice. 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 0.314, it tells us if this model predicts 100 POIs, there would be 41 people are actually POIs and the rest 49 are innocent. With recall score of 0.587, this model finds 45% of all real POIs in prediction. 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.