# Enron POI Identifier Report

*By Henry Wang*

## Introduction

The project explores algorithmic classifiers with the objective of finding persons of interest, POI, from public Enron financial and email data.

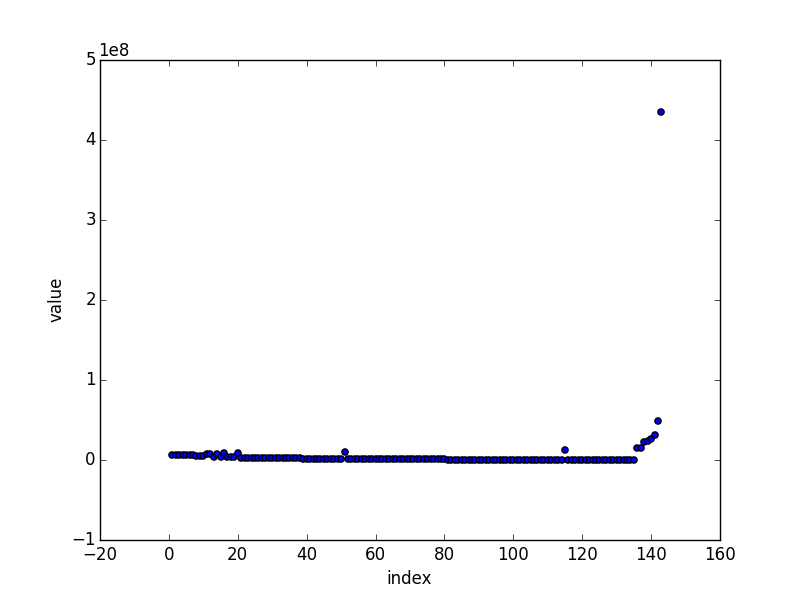
## The Enron Data

The data contains 20 features on 145 persons with a subset of 18 marked with the additional POI flag. POI are those who were indicted, reached a settlement, involved in plea deals with the government, or testified in exchange for prosecution immunity. Some features did not contain data for every person.

Various supervised machine-learning algorithms are explored with this data to find the optimal classifier for the POI flag based on selected features.

## Removing Outlier Data

There was one obvious outlier in the data—the total combined values for all persons. The figure below shows the data with outlier at the top of the chart for the feature “total\_payments.” This outlier was removed for testing.



### Feature Transformation

All features are scaled from 0.0 to 1.0 for testing. This is done for some classifier, such as SVM, where the scales mattered.

Three additional features are added to describe data in terms of ratio to overall amount. The ratios help make better comparisons over absolute values.

* fraction\_to\_poi: ratio of emails sent to POI
* fraction\_to\_shared\_with\_poi: ratio of emails received shared with POI
* fraction\_from\_poi: ratio of emails from POI

### Feature Selection

To select features, two processes were tried: PCA eigenvalues and decision tree importance. The final features are the normalized values are:

* total\_payments: added from PCA analysis for eigenvalue
* loan\_advances: added from PCA analysis for eigenvalue
* total\_stock\_value: added from PCA analysis for eigenvalue
* exercise\_stock\_options: added from PCA analysis for eigenvalue
* fraction\_to\_shared\_with\_poi: added from DT analysis for importance

**PCA**

The eigenvalues for all features of the first PCA component is analyzed. Four features with strong eigenvalues are selected for final selection:

|  |  |
| --- | --- |
| Feature | Eigenvalue |
| salary | -0.008248 |
| deferral\_payments | -0.006891 |
| total\_payments | **-0.681722** |
| loan\_advances | **-0.513378** |
| bonus | -0.05344 |
| restricted\_stock\_deferred | -0.005002 |
| deferred\_income | 0.006889 |
| total\_stock\_value | **-0.405058** |
| expenses | -0.000565 |
| exercised\_stock\_options | **-0.292935** |
| other | -0.073756 |
| long\_term\_incentive | -0.027618 |
| restricted\_stock | -0.11211 |
| director\_fees | 0.000167 |
| fraction\_from\_poi | 0 |
| fraction\_to\_poi | 0 |
| fraction\_to\_shared\_with\_poi | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature 1 | Feature 2 | Feature 3 | Feature 4 | Score |
| total\_payments | total\_stock\_value | exercised\_stock\_options | fraction\_to\_shared\_with\_poi | 1.162 |
| loan\_advances | exercised\_stock\_options | long\_term\_incentive | fraction\_to\_shared\_with\_poi | 1.156 |
| exercised\_stock\_options | long\_term\_incentive | fraction\_to\_shared\_with\_poi | - | 1.155 |
| total\_payments | restricted\_stock\_deferred | exercised\_stock\_options | fraction\_to\_shared\_with\_poi | 1.133 |
| total\_payments | expenses | exercised\_stock\_options | fraction\_to\_shared\_with\_poi | 1.126 |
| total\_payments | exercised\_stock\_options | director\_fees | fraction\_to\_shared\_with\_poi | 1.116 |
| total\_payments | exercised\_stock\_options | fraction\_to\_shared\_with\_poi | - | 1.115 |
| total\_payments | loan\_advances | exercised\_stock\_options | fraction\_to\_shared\_with\_poi | 1.114 |
| deferral\_payments | total\_payments | exercised\_stock\_options | fraction\_to\_shared\_with\_poi | 1.111 |
| total\_payments | exercised\_stock\_options | long\_term\_incentive | fraction\_to\_shared\_with\_poi | 1.109 |

**Decision Tree**

A decision tree is analyzed with all possible combination of up to 4 features. 3,206 different combinations were tried. The top ten combinations of features are:

For the top ten performing combinations of features, the top sums of importances for feature are recorded.

* exercised\_stock\_options, 5.4
* **fraction\_to\_shared\_with\_poi, 2.8**
* total\_payments, 1.3
* long\_term\_incentive, 0.4
* total\_stock\_value, 0.1
* loan\_advances, 0.0

The fraction\_to\_shared\_with\_poi feature is a feature not present in the PCA analysis. It is added to the final feature selection. The rest are either already included from PCA analysis or not included due to low importance.

The fraction\_to\_shared\_with\_poi feature is a custom feature added to the original set of features. Because it is part of the feature list for top 10 performing combination of all possible features (original and additional), without it performance will not be as high.

### Algorithm Selection and Tuning

Multiple algorithms and parameters are tested:

* Naïve Bayes
* Decision Tree with min\_sample\_split of 1, 2, 5, 10, 15, 20, 25, 30
* SVM of linear/rbf kernel and C-value of 10, 100, 1000, 10000
* AdaBoost with Decision Tree of 2, 10, 20, 30 min\_sample\_split
* Random Forest with Decision Tree of 10, 25, 50, 100 n\_estimators and 20 min\_sample\_split

Tuning the parameters of algorithms means adjusting possible settings to find the best settings. If not done well, the best algorithm may be overlooked. The list above includes tuned parameter values.

Score are recorded for each trial. The score is determined by the sum of precision and recall where both values are above 0.3.

The results of the top 10 best scores are listed below.

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Precision | Recall | Total |
| DecisionTree min\_samples\_split=30 | 0.614 | 0.558 | 1.172 |
| DecisionTree min\_samples\_split=15 | 0.614 | 0.556 | 1.17 |
| DecisionTree min\_samples\_split=20 | 0.614 | 0.555 | 1.169 |
| DecisionTree min\_samples\_split=25 | 0.609 | 0.556 | 1.165 |
| DecisionTree min\_samples\_split=10 | 0.606 | 0.558 | 1.164 |
| DecisionTree min\_samples\_split=2 | 0.48 | 0.418 | 0.898 |
| AdaBoost min\_samples\_split=20, n\_estimators=50 | 0.482 | 0.411 | 0.894 |
| DecisionTree min\_samples\_split=1 | 0.465 | 0.412 | 0.878 |
| DecisionTree min\_samples\_split=5 | 0.467 | 0.408 | 0.875 |
| AdaBoost min\_samples\_split=2, n\_estimators=50 | 0.519 | 0.355 | 0.874 |

### Decision Algorithm Tuning

To further tune the decision tree (see task\_7.py), min\_sample\_split values of 5 to 50 are tested with increments of 5. Each trial is averaged over 50 runs. The table below shows the result.

The best decision tree of min\_sample\_split of 30 is saved in my\_classifier.pkl.

|  |  |  |  |
| --- | --- | --- | --- |
| Split | Precision | Recall | Total |
| 5 | 0.46891 | 0.40918 | 0.87809 |
| 10 | 0.606512 | 0.55727 | 1.163782 |
| 15 | 0.609789 | 0.5568 | 1.166589 |
| 20 | 0.613056 | 0.55654 | 1.169596 |
| 25 | 0.612996 | 0.55654 | 1.169536 |
| 30 | **0.613018** | **0.55665** | **1.169668** |
| 35 | 0.612907 | 0.55057 | 1.163477 |
| 40 | 0.576708 | 0.37515 | 0.951858 |
| 45 | 0.540263 | 0.27694 | 0.817203 |
| 50 | 0.545138 | 0.27541 | 0.820548 |

## Validation

Validation ensures that the classifiers are not over fitting data. Splitting the data into a training set and a testing set can achieve this. Also, multi-fold cross validation can be used.

In this project, 10-fold cross validation is used to divide the data into multiple training and testing sets.

A classic mistake is to not shuffle the data before dividing it into a training subset and testing subset. In this project, data is shuffled before validation.

## Conclusion

The evaluation score used is the sum of precision and recall where both values are above 0.3. Recall measures correctly predicting POI provided that the actual person is POI. The precision measures correctly predicting POI from all predicted (correct and incorrect) POI.

The best classifier of decision tree with min\_split\_split 30 has a precision of 0.613 and recall of 0.557 for a combined score of 1.170.