Identifying Person of Interest in Enron Fruad Case

Jenny Hung

Project Goal

We analyzed the Enron data set, which composes of 20 finance-related and email-related variables for 146 Enron-associated persons. Our goal is to help identify all POIs from the available Enron data. There are many missing values – the exploration shows that the dataset has 3066 data points, of which 1708 data points are valid. The dataset is unbalanced in that there are only 18 POIs v. 128 Non-POIs. Further, The outlier identifier captured two types of outliers: 1) invalid name for a person: "TOTAL" (a summary of all persons) and "THE TRAVEL AGENCY IN THE PARK" (not a valid name); 2) invalid data points: "LOCKHART EUGENE E" (contains over 95% invalid values). The outliers are removed before proceeding with the data analysis.

Features Used

New features were engineered from the original features by first constructing the finance- and email-ratios by dividing each of them by 'salary' and 'from_messages', then sum them up. The new features (and final features list) are therefore: 'finAll', 'emailAll', 'from_poi_ratio'. More concretely, the new feature 'finAll' can be written as $\sum_{\{i \in finance\ features\}} \frac{salary}{feature_i}$, and $\sum_{\{i \in email\ features\}} \frac{feature_i}{from\_messages}$, provided that the constituent ratios can be evaluated (otherwise it is set to zero).

Additionally, after obtaining the new features, careful inspection revealed that a judicious choice must be made to further process the features. This is because the finAll feature tend to dominate the feature set and overwhelm the other features during training. I have chosen to use Normalizer() with L2 norm to for this purpose.

The motivation for summing the finance- and email-related constituent ratios is two-fold: 1. Not only the original features center around remunerations and communication intensities, I conjected they communicated much more than that - they communicated **the spread of one's remunerations around one's salary** in the finance-related case, and in the email-relate case, **how one's communication pattern around one's number of emails received**. Hence we constructed the finance- and email-related ratios (note that such a transformation is not achievable by PCA). 2. I believe the original dimensions were too high for a relatively small number of data points. Some further consolidation is required considering the number of valid data points we have is required. Hence we sum the finance- and email-related ratios.

To present the effect of the new, constructed features in the learning algorithm, I present the intermediate metrics in Table One. This table compares the precision, recall and $f1$ score obtained by first running the same algorithm with the original features.

Table 1: Comparison of Original Features and Constructed Features

	Precision	Recall	F1
Original Features	0.12823	0.55550	0.20836

New Features (with Decision Tree)	0. 73376	0.23150	0.35196

Models/Algorithms Used

Several models were fitted: a model with decision tree classifier, with support vector machine, AdaBoost with Decision Tree classifier, KBest with Decision Tree, and SelectPercentile with AdaBoost and Decision Tree. Only the model with decision tree classifier returned satisfactory results. Please see Evaluation Metrics for model summary.

Hyper-Parameter Tuning

Many of the algorithms I have tried need to be tuned by parameters. If an algorithm that requires tuning was used without proper tuning, we could get a result that is far worse than not using this algorithm at all. To tune and determine the final parameter values, I have used a max depth of 2 (for decision tree) and a list of values for C and γ (for SVM), and adopted cross validation to pick the best values that maximizes the $f1$ score. GridSearchCV was used to facilitate the tuning and parameter selection in all cases.

Validation

Validation refers to separating the data into training and test sets, where the test sets are used for testing of model performances only. A simple-minded mistake could happen where one trains the model on all the available data (especially when the quantity of data is limited). A good approach, apart from the separation of training and test sets, is cross-validation, which we use in this project.

Evaluation Metrics

My final model using Decision Tree algorithm produces reasonable results (see Table Two below). With a recall of 0.433, and precision of 0.497, It means that there is a POI in the data set, my algorithm will pick it out half the time. It also means that a person is truly a POI approximately half the time given that he or she has been identified as a POI by my algorithm. Comparable metrics for all other experimented models are also reported.

Table Two: Comparison of Evaluation Metrics

	Precision	Recall	F1
Decision Tree	0.49799	0.43300	0.46323
Support Vector Machine	0. 73376	0.23150	0.35196
AdaBoost with DecisionTree	0.41328	0.34550	0.37636
KBest (2) with AdaBoost(DecisionTree)	0.23982	0.19150	0.21296
SelectPercentile (90)	0.29155	0.35350	0.31955

with DecisionTree			