Identity Fraud From Enron Email

By Matthew Murphy

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? [relevant rubric items: “data exploration”, “outlier investigation”]**

The goal of this project is to determine if we can create a model, a Person of Interest (POI) identifier, using the Enron email and financial dataset that will accurately predict if a person should be a POI for fraud. Machine learning can help us create this model and apply it to other data sets.

The dataset contains information that resulted from the Enron scandal in the early 2000’. The dataset contains 146 records with 14 financial features, 6 email features, and 1 labeled feature (POI). Of the 146 records, 18 were labeled as persons of interest. My data analysis led me to remove 2 records, TOTAL (not actual observation) and THE TRAVEL AGENCY IN THE PARK (not an individual).

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. (You do not necessarily have to use it in the final analysis, only engineer and test it.) In your feature selection step, if you used an algorithm like a decision tree, please also give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values. [relevant rubric items: “create new features”, “intelligently select features”, “properly scale features”]**

For feature selection, I used scikit-learn’s SelectKBest model. For me, this had the best precision, recall, and performance. Features were not manually selected. Train test sets were created with Stratified Shuffle Split cross validation due to a small sample size . The features used are:

```
features: salary score: 5.921123
features: bonus score: 8.260053
features: deferred_income score: 4.737779
features: total_stock_value score: 19.415782
features: exercised_stock_options score: 24.735437
features: to_messages score: 2.196165
```

```
features: from_poi_to_this_person score: 1.696272
features: from_this_person_to_poi score: 3.726112
features: shared_receipt_with_poi score: 6.862862
features: to_poi_fraction score: 4.331890
```

Two of these features were engineered for testing.

- to_poi_fraction - fraction of total to emails sent to a POI
- from_poi_fraction - fraction of the total from emails received from a POI

The to and from features help establish a relationship between POIs and other employees. These relationships or links can be important to the identification process.

What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms? [relevant rubric item: "pick an algorithm"]

The model uses the AdaBoost and DecisionTree algorithm because it provides the best results. The other algorithms tried were Decision Tree (no AdaBoost), Random Forest and GaussianNB, and Logistic Regression. Each of these these worked well for some aspects, but not as well at others. As an example, Random Forest provided the best accuracy score but had low precision and recall scores.

3. **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? What parameters did you tune? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly explain how you would have done it for the model that was not your final choice or a different model that does utilize parameter tuning, e.g. a decision tree classifier). [relevant rubric items: "discuss parameter tuning", "tune the algorithm"]**

Tuning is the process of changing, testing, and updating parameters so they are optizmized to produce the best results. If this is not done well, it can lead to false data. An example of poor tuning was I used precision, recall, and F1 score for validation, but mistakenly used accuracy as a benchmark for tuning. This caused issues with my data that did not appear to make sense.

GridSearchCV was used to determine the optimum parameters. GridSearchCV will evaluate combinations and return a classifier that provides the best score.

4. **What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis? [relevant rubric items: "discuss validation", "validation strategy"]**

Validation ensures that a machine learning algorithm generalizes well. A common mistake of overfitting, where the model is trained and performs very well on the training dataset, but markedly worse on the cross-validation and test datasets.

Stratified Shuffle Split cross validation was used to create multiple train test sets. This was an ideal approach given the small sample size.

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. [relevant rubric item: "usage of evaluation metrics"]

Precision, Recall, and F1 scores are used to evaluate the success of predicting POIs.

Precision: 0.43070
Recall: 0.40550
F1: 0.41772

Precision and recall come in at 43% and 41% respectively. This means that of all the people classified as POIs by the model, 43% are actual POIs.   The model correctly identifies 41% of the total POIs. Effectively, the model will capture over 41% of all POIs.