**Enron Submission Free-Response Questions**

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 source dataset is the enron emails corpus which contains emails from Enron and financial data by person. The objective of the project is to go through emails and financials of those persons and try to identify Person Of Interest (POI). A POI is someone who was charge, convicted or link (settlement or testification) to fraud. Going through this amount of data manually would be very long, problematic and prone to error. Using Machine Learning, we can read those data and try to infer behavior from it.

As we want to identify POI, we need to select the right email and financial features to get the best results possible. Going through the dataset, we can see that a TOTAL is present. This is an omission and is removed by hand. I also removed ‘THE TRAVEL AGENCY IN THE PARK’ which was a good indicator but is not a person. All persons were also check for emptiness conducting to the removal of LOCKHART EUGENE E.

In the end, 143 data points were used.

1. 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 doesn’t 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.)  If you used an algorithm like a decision tree, please also give the feature importances of the features that you use.  [relevant rubric items: “create new features”, “properly scale features”, “intelligently select feature”]

The features used are: 'deferral\_payments', 'restricted\_stock\_deferred', 'deferred\_income', 'director\_fees', 'exercised\_stock\_options', 'shared\_receipt\_with\_poi', 'deferred\_stock\_ratio'

I tried several ones, basically removing one at a time and then checking new results.

I created the following new features, but did not ended up using them:

* + Exercised stock ratio = exerciced stock/total stock. I was thinking that fraudulent people would have exercised their stocks when knowing that everything was over
  + Sent to POI ratio: message to POI/total messages sent. The idea here was that POI would send more messages to POI

As my chosen algorithm is RandomForectClassifier, features scaling is not necessary. Nevertheless, during algorithm testing, I also used PCA and RandomizedPCA to see if dimensionality reduction could help (it did not). In the end, I used KernelPCA to get 2 principal components (data points are not linear) and added it to my seven features. When checking for importance, both principal components were at 0. I removed them from the dataset to speeds things up.

The features importance are : [0.01910869 0.0211549 0.24162027 0.09343623

0.19434074 0.39993132 0.03040784].

1. What algorithm did you end up using?  What other one(s) did you try? [relevant rubric item: “pick an algorithm”]

I used a randomforestclassifier. NBClassifier, as the default one gave low scores, but still a baseline. I then tried a SVMClassifier, but ended up with some errors because of the scarcity of positive instances in the dataset. DecisonTree gave with good results but needed to be strengthen. So I tried AdaBoost and RandomForest. Adaboost gave better recall but RandomForest seemed more consistent and balanced between precision and recall so I kept it and worked on getting the best I could from it.

1. 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?  (Some algorithms don’t 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 if you used, say, a decision tree classifier). [relevant rubric item: “tune the algorithm”]

Algorithms are mathematical formula. Tuning a parameter is basically modifying the formula. Parameters can greatly influence scores. For example without tuning, DecisionTree overfit easily and quickly, basically learning the whole dataset.

Using RandomForestClassifier, I first set up a GridSearchCV with several parameters to test:

* + N\_estimators: number of trees in the forest. Tested values were [10, 50, 100, 200]
  + Min\_sample\_split: minimum number of element to split a node; less than 3 implies leaves with only 1 element.
  + Class\_weight: weight of each class. Tested values were [‘auto’, ‘subsample’, ‘None’]
  + Criterion: measure of quality of a split. Tested values were [‘gini’, ‘entropy’]

1. 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 item: “validation strategy”]

Validation is the process to vet a classifier or an estimator generalization capacity. Validation needs to be done against data not used during training as validating against train data is biased (we could only store the labels to the data and do a look up).

Basic strategy is to split the dataset into training and testing sets. I used multi folds and split of the dataset to validate the results and the parameters. Folding allows for the generation of several training and testing sets from the same initial dataset

1. 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”]

Several metrics exist to validate the results. Accuracy is the number of correct classification (poi or not poi in our case) divided by the number of data points.

Recall is the number of POI divided by the number of POI classified by the RandomForestClassifier.

Precision is the number of POI identified divided by the number of correct classifications.

I was mainly concerned with accuracy and recall as I wanted to identify as many POI as possible, the idea being that the list would be given to someone to investigate them properly.

My results on those metrics are:

Accuracy: 0.76907 Precision: 0.34751 Recall: 0.70250

Accuracy of 0.7691 means that 76.91% of the classifications were correct.

Precision of 0.3475 means that 34.75% of the person classified as POI are actually POI.

Recall of 0.7025 means that 70.25% of all POI were identified as such.