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 identify persons of interest in the Enron fraud using a given dataset. This is a rich dataset because it has thousands of emails, tens of people and tens of features that we can use to narrow down, who might have been involved in the Enron fraud. This data set includes 146 items, 18 persons of interest and 127 not persons of interest. Machine learning is useful in this situation because machine learning can speed up the process of identifying these people when us humans give “the machine learning process” a number of features. And thus given training features and labels, machine learning algorithms can group data that are closely related to each other. Machine learning in other words, can not only speed up the process but make it simpler to identify who may be a person of interest. If there were any missing values or values that had “NaN” I simply skipped over that feature for that person.

The main outlier I removed was the “TOTAL” feature in the financial features. I removed that because using my “human intuition” I know that, that’s not a specific person and that’s just a mathematical value we don’t need. I didn’t remove other outliers such as people because in this case it makes sense that people that are outliers may just be the person of interest.

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 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”, “properly scale features”, “intelligently select feature”]

The 5 features I used were salary, exercised stock options, from poi to this person, expenses and assets total. For the most part I used my human intuition to pick the features. I did not do any scaling because I was already getting a high accuracy score of 0.86, precision of 0.48 and recall of 0.37 using AdaBoost. The accuracy, precision and recall for DecisionTreeClassifier was about the same. I also made my own feature called “assets\_total”, which takes the sum of total\_stock\_value, deferred\_income, long\_term\_incentive, salary and bonus. I created this feature because persons of interest are likely to have more money all over the place than other individuals working in the company and even if they have “NaN” values for one of them, we can still add up their other assets and get an approximate value of all their income and assets. And thus it will be easier for the algorithms to train and identify these people because people with higher total assets and that are persons of interest are likely to be more close to each other.

Not surprisingly when I take out the “assets\_total” feature the accuracy for DecisionTreeClassifier drops from 0.83 to 0.80. The accuracy for all 3 algorithms I used (SVM, Adaboost, DecisionTreeClassifer) are all higher with the assets\_total feature. The scores, precision and recall scores are documented at the end of poi\_id.py using DecisionTreeClassifier. I chose these 5 features because they gave me the best accuracy score. I also got a similar accuracy score just using 3 features salary, exercised stock options and from poi to this person but I decided to use 5 features instead to find a “bias-variance balance”.

1. 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”]

I chose DecisisionTreeClassifier because it had a solid accuracy score of 0.833, precision of 0.37 and recall of 0.379. While Adaboost had similar accuracy, precision and recall the training time for Adaboost was much longer, DecisionTreeClassifier trained the same number of data points in 0.001s whereas Adaboost took 1.112s. Parameter tuning means to find a balance between finding a good accuracy score and “good fitting line” to avoid over fitting. For example, in AdaBoost there is a tradeoff between n\_estimators and learning rate. Increasing the n\_estimators to= 450 and learning\_rate to 3.6 will drop accuracy score. I found when I increased learning rate too much my accuracy score dropped. In SVM, C and gamma are two parameters to use to avoid over fitting and find a good accuracy score. I did find that tuning the algorithms such as for Adaboost gives a better accuracy score.

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 splitting the data into training and testing data, and fitting the data with different algorithms and tuning different parameters to see which gives the best results. We separate training and testing data so we can get an estimate of performance (accuracy, precision, recall) on an independent dataset. We can also check if we are over fitting data. The training features are used to train the algorithm, test sets are not used for training/fitting the algorithm. The test set is more of a validation tool to check the steps we are taking for training our data. The test set is used by the machine learning algorithm for predicting and to check performance. In order for the algorithm to work correctly we have to use the “features” or “components” found in the training data to predict the test data. If we use test set to train an algorithm we are giving ourselves an “unfair advantage” and the accuracy score will be very low because the algorithm didn’t have the maximum amount of data it can have to train the dataset. Or if we try to check the performance during training with test set we can also be seeing low accuracy scores because we are training with one set of features/labels but testing with another completely different set of features/labels. For this dataset, we used cross validation. Cross validation can find us the perfect splitting point between training and testing data. We want as many training data as possible for learning means and as many testing data possible to check our validation.

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”]

My accuracy for the decision tree classifier was 0.83313. Which means about 83% of the time the classifier was able to predict POI (1 or 0) accurately (same as the true labels). The precision score for the same classifier was 0.37568. Which means of all the times the classifier said it was a person of interest about 37% of the time the classifier was correct.

For this project precision would be how many persons of interest there are (1 - positives) divided by everyone who was truly a person of interest and people that were not persons of interest but were still selected as persons of interest(false positives). In other words out of everyone that was selected as a person of interest how many are truly persons of interest. Recall on the other hand is out of everyone that is truly a person of interest what percent was predicted properly. (True positive / true positive + false negative), false negative – people that are POI’s but were identified as non POI’s. Accuracy score is simply how many times the predictions matched with the true labels over all the labels.