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.
The goal of this project is to develop a method to identify which Enron top executives may have committed fraud based on the data provided. The 21 features in the data include financial features, email features and a person of interest (POI) label. A POI is identified as someone who was indicted, reached a settlement or plea deal with the government, or testified in exchange for prosecution immunity – our indicator of someone who may have committed fraud. Out of the 146 people in the data set, 18 people are positively identified as a POI.

To narrow down the number of features to further investigate, the percent of all people, POIs, and non-POIs with missing values for each feature are calculated. The financial features that have <50% missing for all people and POIs are: salary, total_payments, bonus, total_stock_value, expenses, exercised_stock_options, other, and restricted_stock. Additionally, the deferred_income and long_term_incentive features are included since they are missing for more than 50% of the non-POIs, but less than 50% were missing for POIs. An entry for these features may be an indicator of a POI. The deferred_income feature is one of the top features (see question on feature selection).

In summary, the following financial features are investigated further:

1. salary
2. total_payments
3. bonus
4. deferred_income
5. total_stock_value
6. expenses
7. exercised_stock_options
8. other
9. long_term_incentive
10. restricted_stock.

Were there any outliers in the data when you got it, and how did you handle those?
It was found “TOTAL” is listed as an individual in the data set and is removed. The restricted stocks for BHATNAGAR SANJAY is the negative value found in enron61702insiderpay.pdf. The entry is kept, but fixed. Additionally, using the featureFormat function removes seven people that had zeroes for all features.

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.
Two new features are created, using the email data: fraction_to_poi and fraction_from_poi. The fraction of a person’s emails related to another POI might be more useful than the pure number of e-mails related to a POI since some people may e-mail a lot or very little. The fraction_to_poi feature is one of the top features (see question on feature selection).

What features did you end up using in your POI identifier, and what selection process did you use to pick them?
Features used for the POI identifier are salary, bonus, deferred_income, total_stock_value, expenses, exercised_stock_options, and fraction_to_poi. Three feature selection algorithms (SelectKBest, RandomizedLasso, and RFE) are used to select ten important features. There is some disagreement, but the seven kept are agreed to be amongst the most important features.

Did you have to do any scaling? Why or why not?
Scaling was used on the features prior to feature selection since the values for the features span many orders of magnitudes (10^{-2} to 10^{7}). Scaling allows the proper relative significance of each feature.

If you used an algorithm like a decision tree, please also give the feature importances of the features that you use.
A decision tree was not used to determine the important features.

What algorithm did you end up using? What other one(s) did you try?
The AdaBoost, RandomForest, Bagging, ExtraTrees, AdaBoost with ExtraTrees, and AdaBoost with RandomForest classifiers were tried out of the box. The five features are scaled before the classifiers were tested. The AdaBoost classifier is chosen for tuning it has a high F1 score and the recall and precision scores are already above 0.3.

What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well?
To tune the parameters of an algorithm is to find a set of parameters that optimize an algorithm to a given data set. The parameters can change the complexity of the algorithm. Without tuning, an algorithm might not perform as well as it could since the initially chosen parameters may not be suited for the problem.

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).
Attempts are made at using GridSearchCV for parameter tuning. However, the resulting parameters from this function (using the precision, recall, or f1 score as the performance metric) actually resulted in lower precision and recall scores. Consequently, I created a version of this that loops through the AdaBoost classifier parameters, and calls the test_classifier function (with only 250 folds) used for the final analysis. From the suite of simulations, I choose the set that produces the highest F1 score, given that the precision and recall scores are at least 0.3.

The parameters tuned for AdaBoost are:

1. n_estimators: The maximum number of estimators at which boosting is terminated.
    a. From 50 to 70
2. learning_rate : Learning rate shrinks the contribution of each classifier by learning_rate.
    a. From 1 to 0.9
3. algorithm:
    a. Stayed with default = SAMME.R

The spacing for n_estimators is +/- 5 and for learning_rate is +/- 0.05. Finer values were not used to prevent overfitting. This improved the precision, recall, and F1 scores from (0.49370, 0.333, 0.39773) to (0.52214, 0.36550, 0.43). The base_estimator was not changed since two other base estimators were tested previously.

What is validation, and what's a classic mistake you can make if you do it wrong?
Validation is testing an algorithm to ensure it's doing what is intended. By testing, one can check the performance of the algorithm and if it is overfitting the data. One classic mistake is to train an algorithm on the same data used for testing. This would not be an independent validation and may cause overfitting. Instead, a data set should be split into a training set and a test set.

How did you validate your analysis?
Using the test_classifier function in the tester script, the data was split into training and test sets using the Stratified ShuffleSplit algorithm with 1000 folds. The overall performance of the algorithm was analyzed with testing data.

Give at least 2 evaluation metrics, and your average performance for each of them.
The precision, recall, and F1 scores are performance metrics used since the data is skewed (many more non-POIs than POIs).

The unturned AdaBoost classifier provided the following average statistics:

```
Accuracy: 0.86553      Precision: 0.49370     Recall: 0.33300 F1: 0.39773     F2: 0.35619
True positives:  666   False positives:  683  False negatives: 1334   True negatives: 12317
```

The turned AdaBoost classifier provided the following average statistics:

```
Accuracy: 0.87080      Precision: 0.52214     Recall: 0.36550 F1: 0.43000     F2: 0.38883
True positives:  731   False positives:  669  False negatives: 1269   True negatives: 12331
```

Explain an interpretation of your metrics that says something human-understandable about your algorithm's performance
The tuned AdaBoost classifier has better precision than recall. That means that whenever a POI gets flagged in the test set, it's likely to be a real POI and not a false alarm. However, sometimes real POIs, are missed.