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 individuals who are involved as part of the Enron scandal. The Enron scandal involved the bankruptcy of a large energy corporation due to financial fraud. What makes this dataset interesting as a machine learning exercise is that financial and email data on many individuals including are available. Using features such as salary, email correspondences, bonus, and other compensation one can use machine learning to attempt to classify individuals who committed fraud versus innocent individuals.

The dataset that we are given contain 146 individuals of which 18 are convicted persons of interest. There are 21 features, of which restricted_stock_deferred, loan_advances, director_fees have over 75% with no values. Outliers rows in the dataset include “TOTAL” and “LOCKHART EUGENE E” because the latter has all columns set to NaN except poi. When plotting some of the compensation metrics, there appears to be outliers, however, these outliers often corresponded to persons of interest and I did not remove these individuals since there are only 18 POIs.

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

I created three features: fraction of messages from this person to poi, fraction of messages to this person to poi, and fraction of compensation from stocks. I put these features and all pre-existing features (excluding emails and with NaNs changed to 0) into SelectKBest to do feature selection. Initially I put in all of these features with a parameter for k of 5, the feature importance from this are as follows:

```
'exercised_stock_options', 25.097541528735494
'total_stock_value', 24.467654047526391
'bonus', 21.060001707536589
'salary', 18.575703268041782
'deferred_income', 11.595547659730601
```

I then used the Pipeline feature to search the parameter space for a suitable K. The value that I ended up using for K was 4 which is the same as the top 5 shown above excluding deferred_income

3. 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 tried DecisionTree, GaussianNB, and RandomForestClassifier. I ended up using GaussianNB since that one had the best performance (highest F1-score, above 0.3 in both precision and recall).

4. 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 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 item: "tune the algorithm"]

Parameter tuning of algorithms is important to identify the optimal combination of parameters for a specific dataset. Tuning parameters can increase accuracy and recall, however, finding the optimal parameters could use up more computational time. I used pipeline with GridSearchCV to search the parameter space to identify parameters that performed well.

For this dataset, I tuned K-Best to select the optimal number of features along with tuning the various algorithms that I tested (with the exception of GaussianNB since it has no parameters to tune). When running the decision tree, I tuned the criterion, min sample split, and min samples leaf. The optimal parameter combinations for these three were 'gini' criterion, 3 min samples per leaf, and 4 min samples per split. When running RandomForestClassifier, I tuned the number of estimators and min sample split, the optimal for these were 10 and 5.

5. 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 used to prevent overfitting. To avoid this, I used stratified shuffle split to split data into training and testing sets. I used stratified shuffle split instead of kfold since I wanted overlapping folds since I had very few poi and also wanted keep the percentage of poi constant since it is a low percentage of total dataset.

6. 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"]

I used the following evaluation metrics: precision, recall and f1-score.

Precision is the percentage of true positives in the set of positives, a precision value of 0.5 means that half of the persons of interest chosen by the algorithm are real. Recall is the percentage of persons of interest is found by the algorithm, a recall value of 0.5 means that half of the total persons of interest in the dataset were identified by the algorithm as true positive. The f1-score is the harmonic mean between precision and recall, this is a measure of the tradeoff between precision and recall.

The final model had precision of 0.503, recall of 0.323, and f1-score of 0.393.