Enron Submission Free-Response Questions
A critical part of machine learning is making sense of your analysis process and communicating it to others. The questions below will help us understand your decision-making process and allow us to give feedback on your project. Please answer each question; your answers should be about 1-2 paragraphs per question. If you find yourself writing much more than that, take a step back and see if you can simplify your response!

When your evaluator looks at your responses, he or she will use a specific list of rubric items to assess your answers. Here is the link to that rubric: [**Link**] Each question has one or more specific rubric items associated with it, so before you submit an answer, take a look at that part of the rubric. If your response does not meet expectations for all rubric points, you will be asked to revise and resubmit your project. Make sure that your responses are detailed enough that the evaluator will be able to understand the steps you took and your thought processes as you went through the data analysis.

Once you've submitted your responses, your coach will take a look and may ask a few more focused follow-up questions on one or more of your answers.
We can't wait to see what you've put together for this project!

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"]

This data set includes a list of Enron employees. The 'poi' column is a binary column that states whether they were a person of interest in the Enron case. The rest of the columns are various statistics about each person, and we will be using those statistics to run machine learning algorithms and see if we can get one that can make decent predictions about who is a poi or not.

There were some outliers in the data, and we used an outlier cleaner to cap the top and bottom 1% of the data.

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"]

I decided to use all the numeric features in the POI identifier so that I could max out the classifiers with as much data as possible. I did decide to use scaling, as it seemed like good practice. I did use SelectKBest at first to determine the "best" features, and eliminated some

with values less than 5, but got worse results on the classifier evaluation metrics, so I went back to using all numeric features.

I did create a new feature which I included in the tests – this was bonus / salary. I wanted a number that would express any bonus they made relative to their salary, as people with higher salaries might get higher bonuses.

However, when I tested using the classifiers with this feature, the results were very poor. Therefore, I took this feature out.

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 RandomForest, SVC and AdaBoost because those seemed like good choices to use when we have a binary label. RandomForest and SVC were not able to come up with good recall numbers, although the other numbers were fine. AdaBoost had both recall and precision above .3, so it was chosen.

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? 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"]

The parameters used when running an algorithm can create vastly different results depending on what is chosen. This means that the algorithm can perform poorly if it is not optimized. Therefore, I used a grid search on all three algorithms in order to determine the optimal parameters.

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 items: "discuss validation", "validation strategy"]

I validated all three chosen algorithms by dividing the data into test and training data, and then using the test results to determine various evaluation metrics for each algorithm, including accuracy, recall, precision and f1 score. If you don't look at precision or recall, you might have a high accuracy that is irrelevant.

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"]

The best performance was with AdaBoost using optimal parameters and it was the following:

Accuracy: 0.8409090909090909

Precision: 0.3333333333333333

Recall: 0.4

F1-score: 0.3636363636363636

The accuracy of .84 means that the algorithm was able to make the correct determination 84% of the time.

The precision of .33 means that 33% of the time the algorithm chose someone as a person of interest, they were one.

The recall of .4 means that 40% of the time, the people of interest were discovered by the algorithm.

The f1 score is the harmonic mean of recall and precision. This is used to balance the trade-off between recall and precision by using one of their means.

NOTE:
After making some changes in the code, like changing the imputer settings from ‘mean’ to ‘constant’ we got the following scores for AdaBoost:

Accuracy: 0.8181818181818182

Precision: 0.3333333333333333

Recall: 0.6

F1-score: 0.42857142857142855