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 the project is to use your machine learning skills to investigate the dataset and identify poi’s from Enron who may have committed fraud. The dataset includes financial and email data in a dictionary that you can use to gain valuable insights using machine learning based off of pattern recognition and computational statistics. The project had a total of 13 features (poi is the first) after I added one new feature and there were a total of 143 people with 18 of them considered POI’s.

Feature	Score
exercised_stock_options	24.82
total_stock_value	24.18
bonus	20.79
salary	18.29
deferred_income	11.4
total_payments	8.77
loan_advances	7.18
expenses	6.09
total_to_bonus_ratio	**4.45**
director_fees	2.13
deferred_payments	0.22
restricted_stock_deferred	0.07

I only found 1 financial outlier I wanted to remove and that was the total value which added up all the financial data in each column. There were multiple outliers in the financial data but they all appeared to be verified values of bonuses and salaries of the CEO and the Board of Directors. Eugene E. Lockhart had no financial values in the dataset and The Travel Agency in the Park wasn’t a person so I removed both to clean up 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 started with all the supplied financial features and narrowed them down using SelectKBest/f_classif. Then I tuned SelectKBest by adjusting the K value to get the best F1 score and I settled on K-8. In my final run I used GirdSearchCV to find the best K number for the features and also for the DecisionTree parameters.

After visualizing the data and noticing that there were multiple outliers (legit numbers) due to high salaries, bonuses, etc. I realized that I needed to scale the data in order to have it all standardized.

I made a new feature of the ratio of total payments vs bonus because I figured the people with a higher ratio of bonus to total payments probably would be a poi. It ended up not being the case because it only scored 4.56 and wasn’t used by GridSearchCV.

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 ended up using the decision tree algorithm because if offered the highest f1 score I was able to obtain after tuning. It took a bit to figure out all the parameters but with a little help from the forum and mentors I was able to get a good score. I tried GaussianNB but wasn’t able to get

as high as scores and I tried SVM but I was never able to get much out of that classifier, it just never stopped running I believe because I choose a linear kernel.

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"]

There are various parameters to tune depending on your classifier when trying to get the best scores out of your data for a given problem. If you don't tune the classifier right it can under-fit (high bias and low variance) or over-fit (describes random noise instead of the underlying relationship) giving you problems and bad scores.

Decision tree has multiple parameters to tune including min sample split, max depth, criterion, max leaf nodes min sample leaf, etc. The Gaussian NB classifier has estimated parameters with no real tuning available. I used GridSearchCV in my final run after some research to help tune the criterion, min_sample_split, max_depth, min_sample_split, and max-leaf_nodes in the provided dictionary.

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"].

Validation is holding out part of your data as a test set and using it to validate your testing data. It you don't validate your data with train/test indices you have a tendency to overfit your data and have to good of a score.

I used Stratified Shuffle Split to cross validate my data with training and testing indices due to the small and imbalanced dataset. It allows you to keep the percentage of the target class as close as possible to the complete dataset and makes the validation models robust.

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"].

Recall score is the ratio of true positives divided by true positives + false negatives. It is the ability of the classifier to find all the positive samples. Average score with DTree was .47 before adding the new feature.

Precision score is the ratio of true positives divided by true positives + false positives. It is the ability of the classifier not to label as positive a sample that is negative. Average score was .55 before adding the new feature.

F1 score is the weighted average of the precision and recall score as following: 2 * (precision * recall) divided by (precision + recall). Average score was .50 before adding the new feature.