Note: for a more detailed explanation of data analysis steps and thought process, refer to DAND_P5.html or DAN_P5.ipynb

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 Enron dataset results from the Enron fraud scandal in the early 2000s. After the data was made freely available, it has been widely used as training data in machine learning (ML) applications. The dataset (as curated by Udacity) contains the financial and email data of select employees. The purpose of this project is to build a classifier to predict whether a person was involved in the fraud (called a person of interest – POI). I looked at both the email and financial data in building the classifier.

   The dataset consisted of 146 employees total, 18 identified POIs, and 22 features, for a total of 3088 observations. Note that POIs were only 12.3% of the total employees, making this dataset imbalanced. All features also had a number of missing values:

```
Number of NaN values for each feature:
bonus                          64
deferral_payments             107
deferred_income                97
director_fees                 129
email_address                  35
exercised_stock_options        44
expenses                       51
from_messages                  60
from_poi_to_this_person        60
from_this_person_to_poi        60
loan_advances                 142
long_term_incentive            80
other                          53
poi                             0
restricted_stock               36
restricted_stock_deferred     128
salary                         51
shared_receipt_with_poi        60
to_messages                    60
total_payments                 21
total_stock_value              20
```

   When I looked at the provided data schema, I deduced that the null values for all the financial data was actually zero, so I set the NaNs in the dataset to zero. For the email data, 41% of the data was missing for each feature. Instead of imputing using the mean/median for 41% of the data, I decided to set these values to zero as well. This would have implications in feature selection below, but was a necessary step.

I identified several outliers and errors in the data as well. I removed rows “TOTAL” and “TRAVEL AGENCY IN THE PARK” from the data after investigation of histograms for each feature. I also investigated other outliers in both the financial and email data, but decided to keep them, since they likely represented true data. Finally, I found a discrepancy in the dataset and the provided data schema (pdf) for the financial data for two individuals (BELFUR ROBERT and BHATNAGAR SANJAY). I decided to remove that data from analysis as well.

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 two new features: the ratio of total payments/total compensation and the ratio of emails involving POI/all emails. After looking at the histograms of all features during EDA, I noticed that there was a difference in POI total payments vs. non-POI total payments, so I thought the ratio of total payments/total compensation may be important. Likewise, when I looked at the histograms for the email data I noticed separation between the classes, so I thought an additional ratio for the email data may be important. I ended up not using either feature in the final classifier.

   I scaled all features using standardization in my data analysis pipeline. I did this because I did a feature union between SelectKBest and PCA, and I wanted the input features for SelectKBest and PCA to be the same. I used standardization instead of normalization because standardization would not suppress the effect of outliers like normalization would.

   Finally, I used automated feature selection, SelectKBest, as part of my analysis pipeline. I used GridSearchCV during optimization to choose the number of features from SelectKBest. The feature scores were:

   [(True, 'exercised_stock_options', 24.29484881566545), (True, 'total_stock_value', 23.651174232774022), (True, 'bonus', 20.352782168409945), (True, 'salary', 18.008236859583267), (True, 'deferred_income', 11.184580251839124), (False, 'long_term_incentive', 9.7014319140157106), (False, 'restricted_stock', 8.9601824650818251), (False, 'total_payments', 8.594044043070511), (False, 'shared_receipt_with_poi', 8.2786814027836879), (False, 'loan_advances', 7.0667108613197493), (False, 'expenses', 5.8166004823459376), (False, 'poi_email_ratio', 5.1621999121024418), (False, 'from_poi_to_this_person', 5.0412573786693846), (False, 'other', 4.0704701087327813), (False, 'from_this_person_to_poi', 2.2951831957380029), (False, 'director_fees', 2.1537689442003294), (False, 'to_messages', 1.5784680483230906), (False, 'deferral_payments', 0.2447738510610257), (False, 'from_messages', 0.17884036938016448), (False, 'restricted_stock_deferred', 0.065999756426723316), (False, 'payment_ratio', 0.026269409668090032)]

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 used the random forest classifier algorithm for my classifier. I also tested the Adaboost and NaiveBayes classifiers. Their average performance is as follows:

(average scores)	*Precision*	*Recall*	*F1*
Naïve Bayes	0.81	0.80	0.80
AdaBoost	0.78	0.84	0.81
Random Forest	0.83	0.87	0.82

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"]

   The search space for optimal algorithm performance can be very large, and the optimum parameters for an algorithm can be very dataset dependent. To tune an algorithm is to adjust the parameters of that algorithm, either "by hand" or using a grid-search type algorithm that explores the width and breadth of parameter space. Without parameter tuning and testing, there is a risk of using non-optimal parameters or finding a local minimum of non-optimal solutions in your model. Thus, your model performs non-optimally.

   For each of the classifiers I tested, I used a gridsearch to tune the parameters for feature selection (selectKbest and PCA) as well as the classifier. The results reported above are the result of parameter tuning. For the random forest classifier, I used GridSearchCV to tune the number of estimators and the min_sample_split. I also tuned the number of PCA components and features selected in the feature union.

   {'random_forest__n_estimators': 25, 'feature_selection__select__k': 5, 'random_forest__min_samples_split': 1, 'feature_selection__pca__n_components': 2}

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 of any model is important to avoid over- or under-fitting a model. ML models are used to predict some behavior of the data. Without validation, it is easy to fit a model on all available data and get very high performance, but then when the model is used it does not accurately describe data not in the original analysis. To avoid this, it is common practice to reserve some data from the analysis that is not used to train the model. The model can then be independently validated on this data to rule out over-fitting.

To validate my model, I used stratified shuffle split to split my data into pseudo- training and testing sets. I used the stratified shuffle split instead of the typical train-test split since the two classes (POI and non-POI) were so imbalanced.

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 precision and recall to validate the performance of my model. Precision looks at the ratio of correct positive observations for each class. For this classifier, it describes how often false positives are (not) raised. Recall looks at the true positive rate, in other words, how often did the model predict true POIs and non-POIs vs assigning them to the opposite class. For my models, both precision and recall are very high for the non-POI class, but it performed less well for identifying POI's. This suggests that the model is biased towards identifying those not involved in fraud, as would be expected for a highly imbalanced dataset.