## Question 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 the identify person of interest POI using the publically available emails and financial data from Enron fraud case. To achieve this goal, different machine learning algorithms are applied and tested to reach the optimum degree of accuracy and precision.

The data set contains 146 entries (rows) and 21 features (variables) and only 18 of the 146 data points are labeled as POI. By analyzing the data set, features with NaN values are replaced by zero for the numeric data type and by the median of the column for the email address feature which was dropped by the end because it’s a string data type and had no effect on the data analysis.

By plotting the data frame, an obvious outlier was detected and by running a quick check on its id, I found that was the total of the salary column. I removed it from the data set.

## Question 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”]

Since we are dealing with a relatively small data base, I choose to use all available features but email address since it had a string data type and had no effect on my analysis as said before, furthermore I created two new features I created 2 new features “to\_poi\_ratio” and “from\_pio\_ratio” with the assumption is that people who received/sent a significant amount of emails from/to POI might also have been a POI. Hence, a closer look at the number of emails being sent/received could be helpful in identifying POIs.

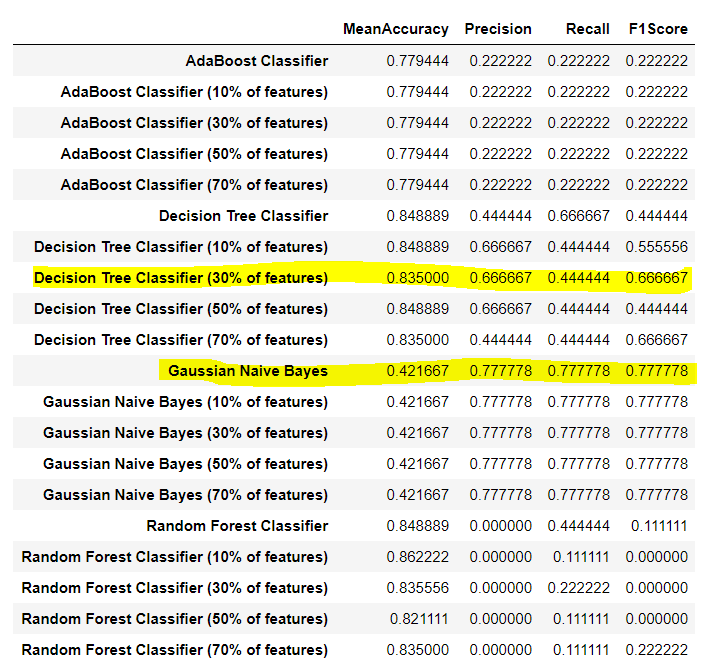
I didn’t do any scaling because I’m not using algorithms that used distance metrics or linear regression in this project it is not necessary to perform scaling on the features. Scaling feature is sweet able for distance metrics and linear regression.

## Question 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 tested the AdaBoost Classifier, Gaussian Naive Bayes, Decision Tree Classifier and Random Forest Classifier for both features list (initial and modified) manually for 10%, 30% 50% and 70% and it seems adding the new features didn’t change the scores for the best classifier (Gaussian Naïve Bayes).

From the table below, it’s clear that the 300% percentile is the best choice for features reduction for the decision tree classifier. Therefor it’s the one used for the final algorithm



## Question 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”]

Tuning the parameters of an algorithm is an important process of machine learning to ameliorate the result by changing the parameters values of the algorithm. Although some algorithms can’t be tuned because it does not have parameters that we need to tune, but most of the used algorithms in the machine learning process are tunable.

If the algorithm parameters are not tuned properly, the result of the analysis may be misleading or biased which will affect the performance and the goal of the machine learning process.

Our number 1 performer, GaussianNB, doesn't actually have any parameters to tune. I choose to tune the parameters of Decision tree classifier in the aim to achieve better performance than naïve base by using the initial data set because this classifier has the second best F1 score.

We will be tuning:

* The minimum number of samples required to be at a leaf node
* The maximum depth of the tree

The process of parameter tuning for a dataset can either by carried out manually by selecting or combining different scenarios or it can be automated by another algorithm such as GridSearch. GridSearch is an approach to parameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid. Since my classifier performance is F1 score, I will tune my algorithm parameter using the min\_samples\_leaf, max\_depth, and criterion.

## Question 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”]

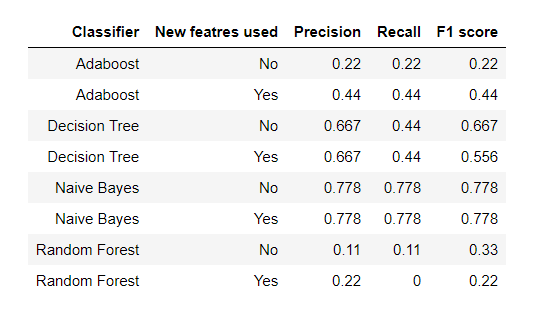
Validation is a process used to evaluate the trained model with a testing dataset. The goal of it is to test the generalization ability of a trained model. Cross validation is a way of measuring the predictive performance of a model. A classic mistake while performing cross validation is overfitting the data. When this occurs a model is trained very well on a set of data but is then not able to make correct predictions in the testing set.

Using StratifiedShuffleSplit returns stratified randomized folds. Since the data set we are dealing with is relatively small and unbalanced, using StratifiedShuffleSplit to randomly split the dataset, and use the whole dataset for both building and assessing the model should lead to a more stable evaluation.

## Question 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”][¶](file:///C:\Users\Hamajid\Desktop\QA\DAND_Enron_Email_Fraud_Identification_Machine_Learning-master\project_5.html#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:-)

By comparing the validated result to the true value ((true positive, true negative, false positive, false negative) I drew a table summarizing the result using both data set as shown below :



\*The metrics evaluation used are precision, recall and f1-score:

* **Precision**: relevant instances among the retrieved instances.

Precision Score is when the algorithm guesses that somebody is a POI, this measures how certain we are that the person really is a POI.

* **Recall**: relevant instances that have been retrieved over the total amount of relevant instances.

Recall, describes the ability of the algorithm to correctly identify a POI provided that the person is a POI.