Enron Fraud Detective Report

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”]

   Project Summary:

   We were given the Enron dataset to performs tasks that would show our learned skills from the machine learning course. The Enron dataset is a dataset containing the financial information and email messages from the Enron company that was involved in a corporate fraud scandal in the early 2000s. This dataset allows us to use our learned skills to create a person of internet list with names of people who played a hand in the fraud.

   Data Summary:

   The Enron dataset contains both financial information and email messages from and to the company’s employees that played a part in the scandal. There were 146 person which shows as 21 rows of data in this dataset. In the dataset there were also 21 features available to use when trying to detect the persons of interest. This dataset has a list of names that can be classified in two categories Person of Interest (POIs) and Non-Person of Interest (Non-POIs). POI is a person that may have played a part in the fraud scandal. Non-POIs being the opposite and not having any sort of indication that they participated in illegal activity (i.e. Plead guilty, indicted, convicted, etc.) from the dataset we were given. There were 18 POIs in the dataset leaving 128 Non-POIs.

   Outliers:

   In the dataset I found the outlier of the ‘Total’ column. This is the adding of each of the datapoints per feature and is unnecessary for the exploration. This creates and outlier with a much might amount than is expected. I removed this altogether using the dictionary’s pop function.

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”]

During my analysis I started by hand picking the features that could be used. After a few tests I decided, however, to add all features available including the newly created features - fraction_from_poi and fraction_to_poi – to the features list. I then started by using feature importance to test the importance of each feature using the DecisionTreeClassifier, which I intended to use. During my first attempt I found this tool very helpful. After my first review the reviewer provided a link and suggested the SelectKBest method. I investigated this option and found that the numbers made a lot more sense. I ended up using the top 7 items from the SelectKbest results as well as the POI feature which was required. As I mention I did complete the fraction on two features and created two new columns as seen below one of the newly created features were used in the final analysis, fraction to POI. I used the first 7 because they were all above the number 10. Originally, I wanted to take only the top 3 with a rating about 20 but I wanted the salary feature to be included in the analysis as I feel this is a key feature to investigate.

```
Select K Best Results:
1.('exercised_stock_options', 25.09754152873549)
2.('total_stock_value', 24.4676540475264)
3.('bonus', 21.06000170753657)
4.('salary', 18.575703268041785)
5.('fraction_to_poi', 16.64170707046899)
6.('deferred_income', 11.5955476597306)
7.('long_term_incentive', 10.072454529369441)
8.('restricted_stock', 9.346700791051488)
9.('total_payments', 8.866721537107772)
10.('shared_receipt_with_poi', 8.746485532129 08)
11.('loan_advances', 7.242730396536018)
12.('expenses', 6.23420114050674)
13.('from_poi_to_this_person', 5.344941523147337)
14.('other', 4.204970858301416)
15.('fraction_from_poi', 3.210761916966744)
16.('from_this_person_to_poi', 2.426508127242878)
17.('director_fees', 2.107655943276091)
18.('to_messages', 1.69882434858085)
19.('deferral_payments', 0.2170589303395084)
20.('from_messages', 0.16416449823428736)
21.('restricted_stock_deferred', 0.06498431172371151)
```

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 all three algorithms that we learned from the course, Naïve Bayes, Decision Tree and Support Vector Machine (SVM). I ended up using DecisionTree to start with because prior to adding StratifiedShuffleSplit it had the smallest difference between recall and precision score when compared with GaussianNB. SVM was not an option for my analysis because the testing time took a lot longer than the other two also the recall and precision scores were the lowest of the three. After adding StratifiedShuffleSplit I noticed the Gaussian NB had the highest scores I have seen while conducting any part of this analysis. A chart is below to show the results of each method.

	GaussianNB	SVM	DecisionTree
Training Time:	0.061 s	0.003 s	0.002 s
Testing Time:	1.054 s	2.772 s	1.068 s
Accuracy Score:	0.85429	0.78307	0.80193
Recall Score:	0.37950	0.14750	0.26850
Precision Score:	0.48716	0.18132	0.29074

4. What does it mean to tune the parameters of an algorithm, and what can happen if you do not do this well? How did you tune the parameters of your 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”]

   Explanation of Tuning Parameters:

   To Tune the parameters of an algorithm is to adjust the specification that are being used in an algorithm so that it will fit the desired outcome the creator is looking for. For instance, with decision tree one might change the maximum depth of the decision tree. When raising this value, the decision tree algorithm would be able to gather more data points and could possibly lead to overfitting depending on other parameters that are set to create the necessary balance needed.

   Parameters Used to Tune My Algorithm:

   In my final submission, using GaussianNB, I did tune any parameters. However, most of my investigation I used the DecisionTree method to try various methods and parameters to tune my algorithm. I used GridSearchCV, MinMaxScalar and the Adabooster option. For the parameters I used best estimator_ for suggestions. I added entropy and gini for the criterion and random state as 42, I also used the random splitter option for my grid search. The provided the algorithm with the best parameters to perform the learning.

5. What is validation, and what is 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 the process of verifying the algorithms ability to correctly predict an outcome. To test the validity of an algorithm there are techniques in the sklearn model selection library that will all for the creation of test sets. These test sets are tested using the algorithm to ensure that the algorithm does not only perform well on the dataset that it was made for but also a random set as well. It is important to do this because another dataset could be introduced with the intention of using this algorithm but if it has not been validated the results could be incorrect.

For my validation I used test_train_split from the sklearn model selection library. This allowed for test sets to be used to validate my algorithm. I also used the StratifiedShuffleSplit as suggested which also allows for splitting the dataset into a train and test set to validate the results. The StratifiedShuffleSplit method considers the classes that are being used in the dataset and their disproportion to one another is applicable.

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"]

   Below are the metrics that were evaluated as well as the counts for predictions and the accuracy score. In the analysis I used the Precision and Recall scores as the two evaluated metrics. In the Analysis I found that 49% of the POIs are identified correctly. 38% of the POIs were identified but not all were actually POIs so essentially the 38% takes into account the false positives. False positives in this analysis would be persons that were labeled POIs but in fact were not one of the POIs.

   Accuracy: 85%

   Precision: 49%

   Recall: 38%

   Total predictions: 14000

   True positives: 759

   False positives: 799

   False negatives: 1241

   True negatives: 11201