**Geoff Noble**

**P5**

**Nano-degree**

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"]**

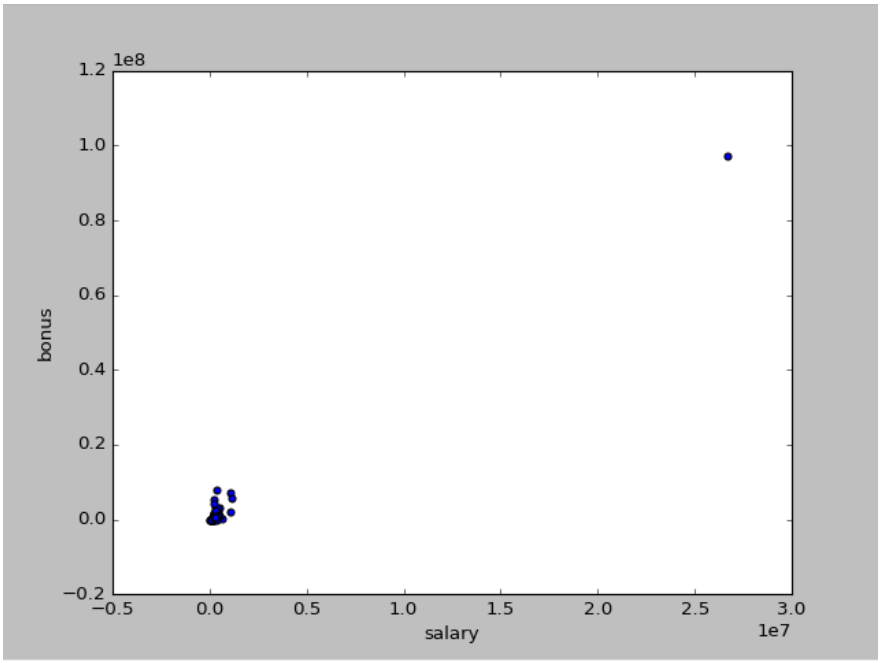
   My goal is to identify people who may have committed fraud at Enron. I can use Machine Learning to study a dataset to unearth patterns that relate to fraud. In this particular dataset (enron_data) I am interested in identifying Persons of Interest (POI). POI's are defined as people who were indicted, testified on condition of immunity or reach a settlement with authorities.

   The project data includes a pickle file which contains the data under investigation. It has 146 records or rows. Each record relates an individual person. Various features such as salary are included in each record. In total there were 21 features.

   The data was spilt into financial and email data. Financial incentives are a seen as common cause of fraud. Outsized incentives (especially those linked to market factors) can affect employee propensity to commit fraud. The email data is generally number of emails messages sent between various recipients. Two of the features include emails to and from people to POIs. This could be useful in measuring collusion.

   In the provided dataset, there are 18 POI's (12% of all records). This is less than the total number of POI's (35) identified poi_names.txt. Some features were missing values. There was not salary information for everyone. This was confined to non-POI's as all POI's has salary information.

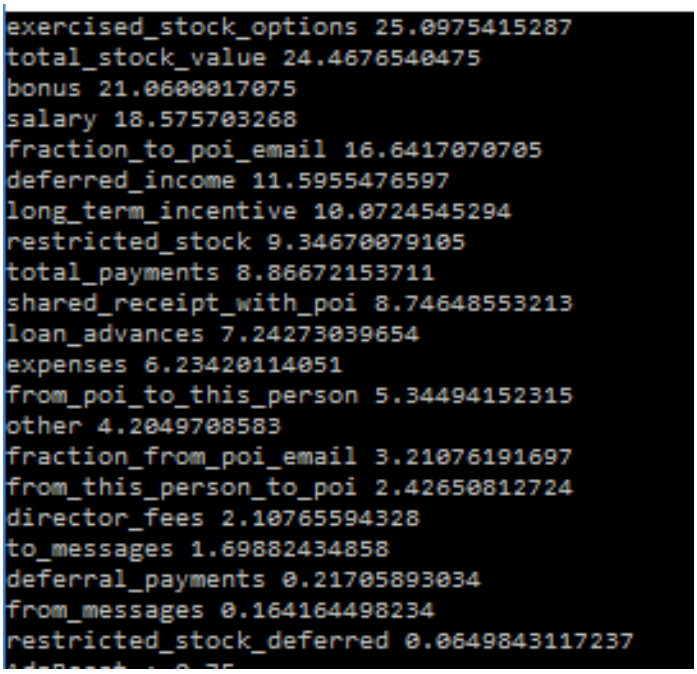
   Next, I needed to test for outliers and potentially remove some of them. I expected to see outliers in financial incentives. In listed corporates, executives at the top tend to receive outsized salaries and bonuses relative to "normal" employees. However, because I am interested in ideas such as outsized financial incentives affecting propensity to commit fraud, I find it unlikely that I would remove outliers in the financial data. I have used a scatter plot to show an outlier below:




I did notice a strange outlier with the label TOTAL which I have removed.

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 used the SelectKBest (Univariate feature selection) method. I produced the following scores:




Based on these scores, I could see that exercised stock options ranked the highest, followed by total stock value. In the end, I chose the top 5 features as ranked above. I tried adding 10 features as a test but this did not improve my precision and recall numbers.

***5 features:***


***10 features:***


I did perform some scaling and used the standard scalar available under preprocessing. Standardization of datasets is a common requirement. Datasets might behave badly if the individual features do not more or less look like standard normally distributed data. In practice we often ignore the shape of the distribution and just transform the data to centre it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation. (Source: sklearn website)

I also created two new features:

- fraction_from_poi_email
- fraction_to_poi_email

I created these as I was interested to see if there was lots of communication between POI's. I would expect a certain level of collusion between POI's and hence above average communication.

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 three algorithms:

- GaussianNB
- AdaBoost
- SVM (abandoned as was way too slow to run)

I tried using the validation from tester.py for these algorithms:

***AdaBoost results:***

```
Precision: 0.29572      Recall: 0.25550 F1: 0.27414
```

***GaussianNB results:***

```
Precision: 0.49545      Recall: 0.32650 F1: 0.39361
```

The metrics were all better for GaussianNB so I chose it.

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 parameters is about improving the performance of your algorithm. It has trade-off, though, as some models suffer from overfitting if tuned too much. I.e. tuning is a process that has diminishing returns. You need to find the optimal about of tuning. You can use manual or automated methods.

I used SelectKBest to tune my parameters and choose the optimal parameters. I also used GridSearchCV for tuning.

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 involves training and testing an algorithm to assess performance and overfitting. Overfitting occurs when the algorithm is fit too closely to the training data. This means that it performs really well on the training data, but poorly on any other new unseen data.

To validate I used the test_classifer function from tester.py (this function uses StratifiedShuffleSplit to split data). The two metrics I concentrated on were precision and recall.

## 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 decided to use recall and precision. Recall is defined as [Correctly classified as POI / (Correctly classified as POI + Incorrectly classified as non-POI)]. i.e. Out of all the items that are POI's, how many were correctly classified as POIs. Or simply, it is the proportion of Correctly classified POIs to total individuals are POIs.

Precision is [Correctly classified as POI / (Correctly classified as POI + Incorrectly classified as POI]. I.e. Out of all the items labelled as POIs, how many are truly POIs. Or simply, it is the proportion of correctly classified POIs to total individuals that were classified as POIs

Precision and Recall for my chosen algorithm were calculated as follows: