Ana Javed
P5 Write-Up

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 was to analyze emails obtained from employees at Enron after a large scale of fraud occurred in the company with machine learning algorithms. With the chosen algorithms, it was encouraged to optimize the parameters to increase the recall and precision evaluation metrics. This would ensure your algorithm had a higher chance of correctly identifying emails belonging to "person of interest" (POI), or employees who were involved in the fraud.

Machine learning was useful in accomplish this goal because it allowed for certain features of the data to be selected and interpreted, for example total stock value owned by an employee, and number of emails from a POI to an individual. There were three outliers identified in the data. Two of the outliers were not employees ("TOTAL" and "THE TRAVEL AGENCY IN THE PARK") and were found by thoroughly examining the list of employee names. The final outlier ("LOCKHART EUGENE E") was found with python code that searched for entries with "NaN" values for its features. The outliers were removed before training the algorithm to prevent errors in predications. After the outliers taken out, there were about 143 data points remaining. Of the 21 features for each individual, 12 were selected for the analysis. Fields that had "NaN" values, such as the "restricted_stock_deferred" or "loan_advances" features, were replaced with zeros in the dataset. Furthermore, there were 18 total POI in 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", "properly scale features", "intelligently select feature"]*

I created two new features "ss_ratio" and "sb_ratio". I took the ratio of salary and total stock value to make "ss_ratio" for each employee. The "sb_ratio" is similar, as it's the ratio of bonus to salary value per employee. Examining ratios were significant to properly measure the wealth an employee had gained while working at Enron. A POI is more likely to have

higher stocks and investments in the company than an non-POI, and thus higher values for the “ss_ratio” and “sb_ratio” features. A POI could also commit fraud with stocks and bonuses, so I wanted to explore them further. In the end I did not include these features in my POI identifier because they ultimately did not increase the accuracy, precision, and recall scores of the algorithm overall.

The twelve features used in the analysis were ‘poi’, ‘salary’, ‘total_payments', 'bonus', 'restricted_stock_deferred', 'deferred_income', 'total_stock_value', 'exercised_stock_options', 'long_term_incentive', 'restricted_stock', 'shared_receipt_with_poi', 'from_poi_to_this_person'. They were chosen by hand, and the list was modified accordingly after determining which features had a greater influence on the classifier than others. Scaling was also conducted in the pre-processing stage of the project. I wanted the selected features to have equal weight when the algorithm was learning, and I utilized Min Max Scaler to accomplish this.

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”]*

After testing different algorithms on the dataset, I completed the project with the Naive Bayes classifier. Other algorithms examined were SVM and Decision Trees. With Decision Trees I was not able to use the Min Max Scaler, and despite utilizing GridSearchCV to optimize the classifier parameters, my evaluation metrics (such as recall and precision scores) were still low. While I was able to use the scaler with SVM and could produce higher recall and precision values than Decision Trees, it was not as high as the values obtained by Naive Bayes.

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”]*

To tune the parameters means to optimize an algorithms learning on a given dataset. If you do not do this well, you risk the chance of over fitting the data. In addition, the trade off between variance and bias of an algorithm is imbalanced with incorrect tuning. For the Naive Bayes classifier, I implemented GridSearchCV for the PCA parameters (specifically “n_components” and “whiten”), but did not optimize parameters of Naive Bayes. GridSearchCV can also be applied to other classifiers with parameters such as SVM.

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 the process of checking your algorithm on an independent set of values to measure its performance. This is a critical step to ensure that over fitting has not occurred. A few mistakes that can be made are training the algorithm on the testing set, making predictions on the training set, or inadequately splitting the testing and training data. I validated my analysis with sklearn's cross validation and StratifiedShuffleSplit() which assisted me to properly maximize accuracy on a small sized dataset.

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"]

Two evaluation metrics conducted were recall and precision. My analysis produced the results 0.39150 and 0.41782 respectively for the metrics. In essence, the recall score (ranging between 0 and 1) displays the ability to "find all the positive samples", or properly detect all true POI in the Enron dataset. Precision, in contrast, displays the ability to "not label positive samples as negative", or to not mistake a POI as a non-POI. Precision scores also range between 0 and 1. In my analysis, my precision value is higher than recall, which means it does a good job of identifying false positives but is weaker in determining false negatives.

Sources:

- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html