Enron Submission Free-Response Questions

1. The goal of this project is to identify people who were involved in the Enron corporate fraud that took place during the late 1990's and early 2000's. This project will attempt to identify the employees that were directly involved in the fraud, also known as person of interest (POI), and the employees that were innocent (non-POI).

   Machine learning can be used to identify POI's through supervised learning algorithms. This dataset contains email and financial information of 145 employees. Among the information is a label that classifies whether an employee is a POI. These features and labels can train a supervised machine-learning algorithm to identify a POI. Other observations:
   - There are 3066 data points; however, 1358 of those data points are missing values.
   - There are 18 POI and 127 non-POI out of the 145 employees.
   - There are 21 features.
   - All of the features are missing some values for all employees. The missing values were converted to zero to avoid distorting the data. The features "long_term_incentive", "director_fees", "restricted_stock_deferred", and "deferral_payments" are missing the most number of values. Features such as "loan_advances" have outliers but those outliers were left in because they tell an important story.
   - The cumulative value of the financial data is in the "TOTAL" row. This row doesn't refer to an employee and all of its values are outliers so it was removed from the dataset.
   - Another metric of outliers used was number of missing data in each row. Through this investigation, 5 rows were removed because they were missing data for almost every feature. These 5 rows are: 'WHALEY DAVID A', 'WROBEL BRUCE', 'LOCKHART EUGENE E', 'THE TRAVEL AGENCY IN THE PARK', and 'GRAMM WENDY L'.

   All of the observations above were found by converting the data into a Pandas dataframe and performing EDA.

2. The features chosen for the final analysis project are:
   - salary, score: 15.54
   - bonus, score: 18.24
   - total_stock_value, score: 21.13
   - deferred_income, score: 12.05
   - exercised_stock_options, score: 21.73

   They were chosen using StratifiedShuffleSplit cross-validation in conjunction with the SelectKBest function in sklearn with parameter f-classif. This parameter was used because the POI data in these features were often the cause of the large variance. Focusing on the features with the largest variance may help identify POI. The above 5 features were chosen after a few rounds of trial and error with the top 10 highest f-scores. Their combination resulted in the highest overall metric scores.

   A custom feature, 'convo_with_poi' was created as part of this analysis. This feature is the proportion of emails sent to a POI out of the total emails sent by each employee. The "to_messages" were chosen because most of the POI's were high-ranking members in Enron and it could be common to receive emails from them. However, if an employee disproportionally sent messages to POI, they might have been involved in the fraud themself. The effect of the 'convo_with_poi' feature on the final algorithm is a 1% decrease in precision.

3. I decided to use Gaussian Naïve Bayes as my final algorithm. It is the simplest ML algorithm with no parameter tuning and quick training time. I tried using SVM with default parameters, SVM with tuned parameters and scaled features using both MinMaxScaler and StandardScaler, and Decision Tree with tuned parameters using GridSearchCV on the latter two. The table below summarizes the performance of each of the tested algorithms.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)
Gaussian NB	86	51	37
SVM (default)	92	0	0
SVM (tuned)	87	75	14
Decision Trees (tuned)	85	47	14

   The above table shows that the Gaussian Naïve Bayes algorithm had the best performance across all three metrics; although, the SVM algorithm performed better in the accuracy and precision metric.

4. Tuning the parameters of an algorithm means adjusting the formulas and calculation methods used. For example, a SVM classifier can use an 'rbf' or 'linear' kernel to transform non-linear data for classification. Parameter tuning is important because the parameters of a ML algorithm have great influence on performance. Tuning parameters can increase accuracy and other metrics. The parameters should be chosen depending on the size, type, class distribution, and other variables of the dataset.

   I used GridSearchCV in the model selection package of sklearn to tune the parameters of the SVM and Decision Tree algorithms. The SVM parameters tuned were: C, kernel, gamma, and class_weight. The class_weight parameter added a slightly higher weighing to the POI class because of the large class imbalance in the dataset. SVM algorithms do not perform well with highly unbalanced classes.

5. Validation is the process of checking the metrics of a ML algorithm. The validation process ensures that results from the ML algorithm can be trusted. Several validation metrics are:
   - Accuracy – proportion of test data correctly classified
   - Recall – proportion of true positives out of true positives and false negatives
   - Precision – proportion of true positives out of true positives and false positives
   - F1-score – weighted average of the recall and precision

   A classic validation mistake is not splitting up the dataset into training and testing subsets. If the entire dataset is used to train a ML algorithm, then the validation will be misleading. All of the metrics will be very high because the algorithm will 'overfit' to that specific dataset and will be able to predict well when tested on the same data. However, the ML algorithm may perform poorly when used on new data because it was adjusted to a specific dataset and cannot handle new information well. In other words, it will have a high bias and low variance. Therefore, it is important to train and test the algorithm on different subsets of the data to determine realistic performance metrics.

I used the ‘StratifiedShuffleSplit’ (SSS) function to validate my algorithm and ensure that it was trained and tested on different subsets of the Enron data. The SSS function creates several different training and testing sets from the dataset with shuffled indices and balanced class distributions from the features and labels. This allows multiple training and testing runs using the same overall dataset, improves the performance of algorithms by supplying an even class distribution, and is very useful for small and unbalanced datasets. The accuracy, recall, and precision scores were measured for each run and the average metrics were shown.

6. The average performances of two metrics from the chosen algorithm are: recall – 37%, precision – 51%. In this specific case, the recall measures the guilty that were classified as innocent (37% of the guilty people were caught) and the precision measures the innocent that were classified as guilty (51% of those who were classified as guilty were actually guilty).

References

- Udacity Intro to Machine Learning Course (www.udacity.com)
- Udacity forums (https://discussions.udacity.com/c/nd002-p5-intro-to-machine-learning/p5-identifying-fraud-from-enron-email)
- Stack Overflow (www.stackoverflow.com)
- sci-kit learn documentation (http://scikit-learn.org/stable/index.html)

“I hereby confirm that this submission is my work. I have cited above the origins of any parts of the submission that were taken from Websites, books, forums, blog posts, github repositories, etc.