Enron Submission Free-Response Questions

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"]*

   In this project, we aim to use machine learning to identify Enron employees who may have committed fraud based on the the Enron financial and email dataset. The dataset itself contains the details of 146 Enron employees, 18 of whom are persons of interest (POIs). Each employee is associated with 21 features, including salary, bonus, total stock value, and so on. From these features, we hope to be able to extract several useful ones, and use them to identify the POIs in the dataset.

   However, the Enron dataset is not perfect. Firstly, many features, especially financial-related ones, have a lot of missing values. Removing observations with missing values is unwise because the full dataset itself is already small. Hence, what I did was to replace all NaN values with zeros. Secondly, there are two outliers in the dataset, namely "Total" and "The Travel Agency in the Park". Since both of them are not Enron employees, these two rows are irrelevant and were removed from the dataset. All other observations, which are Enron employees, are retained in the dataset.

2. *What* ***features did you end up using*** *in your POI identifier, and what s****election 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 decided to engineer two new features, with the following formulas:

   - ratio_email_sent_to_poi = from_this_person_to_poi ÷ from_messages
   - ratio_email_from_poi = from_poi_to_this_person ÷ to_messages

   The rationale is, the number of sent and received emails varies a lot among different employees. For example, it is not fair to compare the numbers of email between a senior director and a lower-level manager. A ratio would better illustrate the significance of someone's relationship with a POI.

   Next, I did a scaling of features using MinMaxScaler. This is because the magnitude of different features ranges widely from less than one for ratio features to millions of dollars for bonus and salary. Then, I employed SelectKBest algorithm to pick several useful features for machine learning. The number of features selected was determined using GridSearchCV. Below are the list of features and their scores. Five selected features are highlighted in yellow.

Features	Feature Score
Exercised stock options	24.8
Total stock value	24.2
Bonus	20.8
Salary	18.3
Ratio of emails sent to POIs	16.4
Deferred income	11.5
Long-term incentive	9.9
Restricted stock	9.2
Total payments	8.8
Shared receipt with POIs	8.6
Loan advances	7.2
Expenses	6.1
Emails from POI to this person	5.2
Other	4.2
Ratio of emails from POIs	3.1
Emails from this person to POI	2.4
Director fees	2.1
To messages	1.6
Deferral payments	0.2
From messages	0.2
Restricted stock deferred	0.1

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 trying several algorithms, I ended up choosing Naïve Bayes because it performs better than the rest. The algorithms that I tried are as follows:

Model	Precision*	Recall*
Naïve Bayes	0.6	0.6
Support vector machine	0.0	0.0
Decision tree	0.33	0.4
K-nearest neighbors	0.5	0.4
Adaboost	0.5	0.2
Random forest	1.0	0.2

** Precision and recall in this table concern POIs only. The values are based on a simple train-test split (70% training set, 30% testing set).*

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 means adjusting the parameters of an algorithm so that it can handle a particular dataset better. Different parameter settings will result in different decision boundaries. If we don't tune the parameters well, the algorithm won't be able to generalize a dataset well and as such, the final classification result might be less accurate.

While exploring for algorithms, I used GridSearchCV for parameter tuning. For example, for K-nearest neighbors, I tried tuning two parameters: n_neighbors (3, 5, 7, or 9) and weights (uniform or distance). GridSearchCV then explored all possible n_neighbors-weights combination and chose one that generates the best F1 score.

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 way to assess the performance of a machine learning algorithm using the given dataset. This is done by splitting a dataset into training and testing datasets, and comparing our machine learning results with the labels in the testing dataset. A classic mistake is to use all observations available in a dataset to train the machine learning algorithm, and then to end up overfitting the given dataset.

To assess the performance of my final algorithm, I used the Stratified Shuffle Split cross-validation. This is because the Enron dataset is small and unbalanced (many more non-POIs than POIs). Stratified shuffle split cross-validation could construct new instances from the dataset to ensure better representation of POI class in both training and testing datasets. Then, it does a randomized k-fold cross validation on the 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"]*

For my final algorithm choice, Naïve Bayes, the average performance is as follows:

- Precision: 0.426
  This means that out of those identified as POIs by the model, only 42.6% are true POIs.

- Recall: 0.351
  This means that the model is only able to point out 35.1% of POIs accurately.

- F1: 0.384
  This is a weighted average of precision and recall scores. Higher F1 score indicates higher precision and recall scores, and hence a better model.