Enron Submission Free-Response Questions

A critical part of machine learning is making sense of your analysis process, and communicating it to others. The questions below will help us understand your decision-making process and allow us to give feedback on your project. Please answer each question; your answers should be 1-2 paragraphs per question. If you find yourself writing much more than that, take a step back and see if you can simplify your response!

When your coach evaluates your responses, he or she will use a specific list of rubric items to assess your answers. Here is the link to that rubric: Link to the rubric
Each question has one or more specific rubric items associated with it, so before you submit an answer, take a look at that the rubric. If your response does not meet expectations, you will be asked to resubmit.

Once you've submitted your responses, your coach will take a look and ask a few more focused follow-up questions on one or more of your answers.

We can't wait to see what you've put together for this project!

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"]

   Enron had so many employees and only some of them were involved in the scandal. We already know fraction of those who were involved but we don't know all of them so here in this project we are trying to employ machine learning techniques to find out who else were possibly involved in this scandal.
   Total number of data points are: 144
   Allocation across classes (POI/non-POI): (18, 126)
   Total number of features in the raw dataset are: 21
   Total number of features used in this project are: 9

   I could find at least one outlier in the data which was 'TOTAL', and I removed it. Also there was another item whose features were all equal to zero ("LOCKHART EUGENE E"), I removed that as well. Based on my outlier detection method I could find some other candidates for outlier but I decided to keep them as they were known POIs so keeping them may help the classification. To find outliers I simply used scatter plot on different axis of data for example, 'salary' vs 'bonus'; I also used dimensionality reduction method (PCA) to be able to plot data using more information. More specifically I used 'salary' vs first transformed factor of other financial data: some of the data point that were candidate for outliers:

- *Lay, Kenneth*
- *Skilling, Jeffrey*

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 doesn't 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.) If you used an algorithm like a decision tree, please also give the feature importances of the features that you use. [relevant rubric items: "create new features", "properly scale features", "intelligently select feature"]

Since the scale for these two factors are very different, feature scaling seems to be important. However I ended up using RandomForest in which feature scaling doesn't impact much.
(As I mentioned for question 3, I tried SVC as well, for which I used feature scaling)

Also, I defined two new feature for my classification and I used them instead of four contributing factors:
From_poi_to_person / from_messages
From_person_to_poi / to_messages

The reason I used these two new features were it is similar to feature scaling and it makes different features to be in similar scales in classifications but more importantly these are better indication of to what extend a specific person were involved with POIs.

Here is the list of all features that I used:
selected_features_list = ['poi', 'from_poi_prop', 'to_poi_prop', 'shared_receipt_with_poi', 'salary', 'bonus', 'total_stock_value', 'restricted_stock_deferred', 'total_payments']

The first one is the label, so I used 8 different features to train RF algorithm.
feature importances of RandomForestClassifier: [0.18132923 0.17936594 0.09689551 0.00039633 0.17599177 0.17719354 0.07310256 0.11572512]

I used these features mostly by hand, I used different combinations to test out which features work out best and I ended up using the 8 above features.
I did following things to come up with best features:

- *I studied sample of data to understand all the features*
- *I tried different combination of features on train and cv dataset to see what combination gives a better score.*
- *I divided features into three main categories of payment, stock and email info*
- *From each category I chose the ones that worked the best with the score of the CV dataset for my algorithm and also capture the most information. For example, by looking at data one can obviously see that bonus is a good candidate while loan*

advances is not as the former has very wide range of numbers and the latter is NaN for more than 90% of cases!

3. What algorithm did you end up using? What other one(s) did you try? [relevant rubric item: “pick an algorithm”]
I ended up using Random Forest which gave me the best results.

One new method that I used was anomaly detection using multivariate Gaussian distribution in which I used all the features and the algorithm tries to fit the data to multivariate Gaussian distribution and then based on the error term, identifies the anomalies (in our case probable POIs); but the performance score was not good comparing to Random Forest.
In fact, all the performance evaluation metrics that I was using were very low for this method. One of the reason is probably because of the very limited number of data points. Usually for anomaly detection algorithm like what I used, it is very important to have very large number of data points.

I also tried SVC but I couldn’t get a good result on precision and recall.
For SVC, the accuracy was at 91% which is very similar to what I get in RandomForst, however the precision and recall was 0 and this is because my SVC algorithm couldn’t classify pois effectively and for most/all of the case it predicted the labels as 0.

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 don’t 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 if you used, say, a decision tree classifier). [relevant rubric item: “tune the algorithm”]

To tune an algorithm is to set the parameters of an algorithm in a way that you can get the best result out of the algorithm. Even the best algorithm if has not been tuned properly may work very poorly so it is important to know different parameters of machine learning algorithms and tune them properly so in our application it works efficiently.

We can divide data to 3 folds of train, cross validation and test data. We can train our classifier using train data. Tune the parameters using cross validation data set and then finally report the performance using test data.

For my Random Forest Classifier I used grid_searchCV in which I set the parameters to possible options and the train the data using different set of parameters:
parameters = {'n_estimators':[5, 10, 15, 20, 25], 'min_samples_split':[2, 3, 4, 5]}

best edtimator RandomForestClassifier(bootstrap=True, compute_importances=None,
criterion='gini', max_depth=None, max_features='auto',
max_leaf_nodes=None, min_density=None, min_samples_leaf=1,

min_samples_split=2, n_estimators=10, n_jobs=1,
oob_score=False, random_state=42, verbose=0)

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"]

   To separate training and testing data. You shouldn't report on performance of your machine learning techniques using training data. All evaluations should be done using test set. One common mistake is that student use train dataset to report on how well her learning method works. Well, obviously if you train you learning algorithm on train data, you would see relatively a high performance metrics on the train dataset.
   Validation should not be on train data nor on test data. I mean, can't you use test data to influence your algorithm.

   Typical scenario for many machine learning techniques is to divide data to train, cross validation and test dataset. Using train dataset you can train your algorithm, using cross validation dataset you can tune your algorithm and test set is only to report performance.

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"]

Accuracy: it simply says to what extend the predictor could predict the labels correctly. However this metric is not very effective specially for case of skewed data.
Precision: it says what percentage of the data points that has been classified as poi, are really poi.
Recall: it says what percentage of POIs was predicted (classified) correctly by our algorithm.

accuracy of Random Forest 0.909090909091
f1 score is 0.5
precision score is 0.5
recall score is 0.5