**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 about 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 evaluator looks at 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](https://docs.google.com/a/knowlabs.com/document/d/17-JwNQH1aRxtqMkJ6zpCL_68kh5F6uSbDXcJS26vZWY/pub) Each question has one or more specific rubric items associated with it, so before you submit an answer, take a look at that part of the rubric. If your response does not meet expectations for all rubric points, you will be asked to revise and resubmit your project. Make sure that your responses are detailed enough that the evaluator will be able to understand the steps you took and your thought processes as you went through the data analysis.

Once you’ve submitted your responses, your coach will take a look and may 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”]

In this project, I want to identify the people of interest (POI) from the financial data and email data in file “final\_project\_dataset.pkl”. In the study, I need to clean up the data at first. This includes removing “NaN”, and removing outliers. And then I need to transform the financial data using logarithmic operation, to get a more “normal” distribution. Then I need to apply different machine learning algorithms on the data and find the best algorithm which has the highest accuracy.

There are a total number of 146 data points in the dataset. POI is about 12.33%. Non-POI is about 87.67%.

Three features are used. They are “poi”, “total\_payments”, and “total\_stock\_value”. Feature “poi” has no missing values. However feature “total\_payments” has 21 missing values and feature “total\_stock\_value” has 20 missing values.

1. 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. One is called “fraction\_to\_poi” and the other is “fraction\_from\_poi”.

In the dataset, any person has the feature “to\_messages”, which shows the total number of emails this person received. It also has the feature “from\_poi\_to\_this\_person”, which shows the total number of emails this person received from any POI. Some people may receive very few emails and all the emails are from some POI. These people may well be a potential POI. So it is very necessary to calculate the percentage of the email numbers from POI among all the received emails. This is the reason to create the feature “fraction\_to\_poi”.

So does the new feature “fraction\_from\_poi”. It calculates the percentage of the email numbers to POI among all the sent emails.

I end up using “exercised stock options” and “salary” in my POI identifier. I pick up any two different features and run the algorithm. I pick up the two features with the best results. I do need to do the scaling. However the scaling I choose is just “log(1+data)”. This is to ensure the data has the more “normal” distribution.

1. 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”]

The algorithm I ended up using is “manually selecting features + DecisionTree + GridSearchCV”.

I have tried the following three kinds of algorithms:

* PCA + SVC + Pipeline + GridSearchCV
* PCA + DecisionTree + Pipeline + GridSearchCV
* PCA + RandomForest + Pipeline + GridSearchCV
* manually selecting features + DecisionTree + GridSearchCV
* manually selecting features + RandomForest + GridSearchCV

It appears DecisionTree + GridSearchCV is better than others in all other five types of algorithms above. It has the highest accuracy.

1. 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”]

It is for the supervised learning. Use the training data to try several pre-defined values of different parameters and find out which values can produce the most accurate prediction on test data. Otherwise, you get a lot of wrong prediction.

In my case, I choose several values for a parameter to run GridSearchSV and find which value has highest Recall score and Precision score.

1. 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 to split the data into training data and testing data, train the learning algorithm using the training data, and verify the algorithm on the testing data. The common mistake is “overfitting”.

In my analysis, here is what I do:  
1, we first divide the data set into a training and testing set of, say, 70%-30%. On the 70% of the data that is put in the training set we then run the gridSearchCV() to train the model using the StratifiedShuffleSplit().  
2, Within the 'sss', the data is again randomly divided into 90%-10% training-test sets for 1000 iterations. Each iteration trains a model on 90% of the training set and tests on 10% of the original training set.  
3, Then, the model with the parameters that have performed best on average, is then used to generate predictions on the the 30% of the original data set that was set aside for testing.  
4,The predictions of that model are then compared to the actual values of the dependent variable to score the final model.

1. 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”]

One evaluation is done through f1\_score(). The other one is done through my custom\_scorer().

Function f1\_score() generates higher precision score and recall score than my custom\_scorer()

My scoring function custom\_score() is to maximize the minimum of precision and recall, compared with f1\_score(). This helps improve the performance significantly. F1 score is harmonic mean of precision and recall, so it tries to balance the two. It doesn't maximize either, but tries to keep both high and equal to one another. F1 = 2/(1/recall + 1/precision). So I do not choose F1-score. Instead, I choose to use the minimum of the precision and recall.