

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: [\[Link\]](#) 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"]

The goal of this project is to identify Fraud from the emails that were taken from Enron. Most emails wouldn't contain an individual's financial information (salary, bonus) and the POIs that we have were already determined by looking at the financial information. To identify other POIs from the email we need to look at who is emailing who. In that case the information that we need to look at (the features) is 'to_messages', 'from_messages', 'from_this_person_to_poi', 'from_poi_to_this_person' I kept the 'poi' and 'salary' features as they were the features that were provided by the starter code.

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", "intelligently select features", "properly scale features"]

Looking at the data just because someone is emailing a POI or receiving an email from a POI doesn't necessarily mean that they are part of the scandal. To narrow down potential other persons of interests I thought to look at the percentage of emails that were sent and received and compare that to the total number of emails. If most of a persons emails are being sent to or from a POI then the recipient/sender would have a higher chance of being involved in the scandal.

I wrote two new features: `percent_received_from_poi` and `percent_sent_to_poi` to determine who the POIs were emailing and who was emailing the POIs.

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"]

I ran the training and testing features through Naïve Bayes, SVM, Decision Tree and K Nearest Neighbors. The models were all very close to each other with the exception of the Decision Trees classifier and out of the four of them the K Nearest Neighbors had the best score (0.88)

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? What parameters did you tune? (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 items: "discuss parameter tuning", "tune the algorithm"]

Tuning the parameters of an algorithm means to change some of the variables that the data is compared against, number of features, how the data is analyzed and where to draw the fit line, such as SVM's kernel variable, linear was used in class but changing it to rbf could lead to different scores. If you over tune the data you can create new outliers or use outliers that shouldn't be used due to the slope of your regression line.

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 items: "discuss validation", "validation strategy"]

Validation is checking the validity of the accuracy of the test set that is produced by the classifiers and the features that you have chosen. If you validate your information incorrectly then when you add in data, for example, the Face Recognition Data(Eigenfaces), then you will have more false positives and false negatives than accurate data. Our data is validated by running it through cross analysis to determine if the training and testing data scores match.

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"]

The first evaluation metric I used was a Linear Regression, and it only scored a 0.22 which could indicate that comparing the messages sent and received from the POI isn't a good metric for trying to find additional persons of interest.

The other metric that I used to evaluate the code was the Accuracy Score which returned a score of 76.4. This indicates that the data has a 76% chance to correctly chose a POI based on to and from emails.

