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 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 choose and train ML classifier to identify person-of-interest using financial and email data of Enron employees. It's unlikely that a trained classifier will identify POIs absolutely precise but it could help to identify suspects which can be checked further using other techniques.

The data provided has got lot's of NaNs for different features. Distribution of NaNs in the whole data set:

{'salary': 50, 'to_messages': 58, ***'deferral_payments': 106***, 'total_payments': 21, 'long_term_incentive': 79, ***'loan_advances': 141***, 'bonus': 63, 'restricted_stock': 35, ***'restricted_stock_deferred': 127***, 'total_stock_value': 19, 'shared_receipt_with_poi': 58, 'from_poi_to_this_person': 58, 'exercised_stock_options': 43, 'from_messages': 58, 'other': 53, 'from_this_person_to_poi': 58, ***'deferred_income': 96***, 'expenses': 50, 'email_address': 33, ***'director_fees': 128***}

The data set consists of 146 data point. Among them there are 2 data points which are considered as outliers as they don't represent any person: "TOTAL" and "THE TRAVEL AGENCY IN THE PARK". There are also 4 data points with all features equal to 0 or NaN. After removal of outliers the data set is 140 data points. Among the data points left there are 18 POIs and 122 non-POIs.

There quite a few features which has got NaN for most data points. It's unlikely that they are useful for classification.

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

Final features list: ['poi', 'salary', 'bonus', 'total_payments', 'exercised_stock_options', 'ratio_bonus_totalp', 'ratio_exso_totals', 'to2from_poi_to_this_person', 'from2from_this_person_to_poi']

The following features were constructed from source data:
ratio_bonus_totalp: Ratio of bonus to total_payments (to check if bonuses in poi total payments are significantly bigger than for others)
ratio_exso_totals: Ratio of exercised_stock_options to total_stocks (to check if poi exercised significantly more/less options than others)
to2from_poi_to_this_person: to_messages / from_poi_to_this_person (If a person receives lots of emails from pois he/she can be poi too)
from2from_this_person_to_poi: from_messages / from_this_person_to_poi (If a person sends lots of emails to pois he/she can be poi too)

I didn't have to do any feature scaling as features used for construction of new features are in same units. In addition to that I decided to use PCA from the very beginning.

I made an attempt to use SelectKBest. The method produced score for features and none of newly created features have beaten original features. The best were still 4: salary, bonus, total_payment, exercised_stock_options.
Scores: [3.18126374e+06 3.92285015e+07 2.78346836e+08 2.26697103e+08
1.53692445e+03 3.56002466e-02 1.64540378e-01 4.42529436e+00]

3.What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms?

I have tried several different classifiers:
- GaussianNB; Doesn't have much parameters to tune. Couldn't make it get recall > 0.3.
- DecisionTree. Was promissing in the beginning but I couldn't find find parameters to satisfy >0.3 for precision and recall. Changng parameters process was like a see-saw: better precision – worse recall and vice versa.
- RandomForrests (took much longer to train/test than decisiontrees);
- SVC (failed to identify enough true positives);
- AdaBoost. Second best classifier with default parameters. Takes quite long time to fit/test it.
- KnearestNeighbours. This classifier showed the best results for precision and recall. I ended up using it a classifier.

4.*What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well?*

Tune the parameters of a classifier is a process of finding optimal parameters for the best result (score/accuracy/precision/recall).

I've done the following changes to default parameters:
- KNearestNeighbours – increased number of neighbours to 3 from default 5.
- Changing default power parameter for Minkowski metric from 2 to 1 boosted recall by approx 0.04

5.What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis?

Validation is a check of model for over-fitting. A classic mistake is not splitting data set into train and test sets.
I have split data set in two parts. One part was used for training of classifier another part for testing. After fitting I checked score of the model on a test set.

It is possible to also use other validation techniques provided by sklearn like cross-validation, k-fold cross validation, GridSearchCV etc.

6.Give at least 2 evaluation metrics and your average performance for each of them.

Accuracy: 0.87220
Precision: 0.53496 Recall: 0.31750

F1: 0.39849 F2: 0.34560

Accuracy tells us that the classifier marked POI correctly in 80% cases.

Precision value close to 0.5 tells us that number of true positives and false positives are close. That means that there is 50% chance that POI will be marked correctly..

Recall value 0.3 tells us that number of false negative predictions is two times more than true positive predictions. The chance of miss-labelling POI as non-POI is about 60%.