Stefan Borchardt

Enron Submission Free-Response Questions & Answers

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"]

This project deals with the Enron scandal and who was involved in it. During the investigation of the scandal formerly disclosed information became publicly available. Financial and email data is used in this project, in order to get insights how people known to be involved in the scandal might be connected to other leading employees. Machine learning can help with this goal because it can recognize patterns even in datasets that are too large to be handled manually. For instance, there are more than 120 MBs of email.

The dataset, as it was used for this project, contains 21 features of financial data and email metadata for 145 persons. Thirty-four of these are known as persons of interest. Despite a wide range of values, there was one outlier which had to be removed - a sum row probably brought in by a spreadsheet application . Some features frequently have missing values, e.g. Loan Advances and Directors Fees.

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 selected the features by trying different combinations by hand based on my intuition. First, I created a new feature, which represents what share of a person's emails are to or from a person of interest. I tried to capture the connectedness to a POI with that feature, assuming that the communication ratio could be an indicator for that. With a fixed set of features, I tried different classifiers, to be described below, and settled for AdaBoost. Because some of the classifiers require scaling (e.g. support vector machines), I scaled all features using a MinMaxScaler. Then, with a fixed classifier and fixed parameters, I tried different combinations:

Features	**Precision**	**Recall**
salary, poi message ratio	.190	.089
salary, poi message ratio, bonus, exercised stock	.414	.300
salary, poi message ratio, bonus, exercised stock, expenses	.421	.263
salary, bonus, exercised stock	.430	.302
salary, bonus, expenses	.558	.351
salary, bonus, expenses, poi message ratio	.514	.321
bonus, expenses	**.629**	**.435**
expenses	.329	.119
bonus	.494	.267

The features I use are Bonus and Expenses. Even though I use a decision tree, I cannot access the feature importance directly, because of the limitations of the pipeline. From the table above it seems that Bonus is most important.

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"]

Using the course and this overview http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html as a starting point, I tried several classifiers:

Classifier	**Precision**	**Recall**
GaussianNB	**.549**	.202
SVC	0	0
LinearSVC	.509	.064
QuadraticDiscriminantAnalysis	**.547**	.188
DecisionTree	.402	**.438**
KNeighbors	.366	.143
AdaBoost	**.483**	**.442**

I chose AdaBoost, because it has a high recall and good precision. Because I can assume that all POIs are labelled correctly, it is safe to use this algorithm (http://www.cs.columbia.edu/~rocco/Public/mlj9.pdf).

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"]

The goal of tuning the parameters is to maximize a certain score or metric. When choosing features and the classifier I tried to balance precision and recall. Since AdaBoost already tunes a decision tree, I chose only two further parameters to tune: `n_estimators` and `learning_rate`, between which the documentation mentions a trade-off.

I tuned using GridSearchCV and maximized the recall, because it seemed to be harder to increase. That approach also balances precision and recall in this case, with my choice of features and classifier.

Classifier	**Precision**	**Recall**
AdaBoost `learning_rate=0.4, n_estimators=25`	.674	.474

It was not necessary to tune different classifiers, because the decision-tree based algorithms (DecisionTree, AdaBoost) performed much better than all others in terms of recall. Instead, I tuned the GridSearchCV by setting the cross-validation folding level `cv` to 6 after trying various values.

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"]

With validation I check if the insights I obtained from the dataset can probably be generalized to new data. It also helps to avoid overfitting. Validation is done by examining if the

results of analyzing only a part of the dataset are valid for the rest of the data. If the subset of data for training is chosen poorly, i.e. is not representative, the model will not fit the testing data.

I chose a variant of cross-validation, which in multiple iterations trains on one partition of the data and test on the other.

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"]

I use precision and recall. From a given haystack, you want to obtain a strategy for finding all needles (high recall) and only needles (high precision) in haystacks. Because needles might have preferred areas in haystacks, but do not tend to be at the same spot in all haystacks, there is a trade-off between recall and precision, and balancing them is, in general, a good approach.

In the context of this project, a recall of .474 means that I identify about half of the POIs in the dataset correctly. The precision of .674 means that 1 of 3 of the POIs I identified are actually not POIs. .