Bryan Smith - bryantravissmith@gmail.com

Project 4 Analysis

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 develop and tune a supervised classification algorithm to identify persons of interest (POI) in the Enron scandal based on a combination of publically available Enron financial data and email records. The compiled data set contains information for 144 people employed at Enron, a ‘Travel Agency in the Park’, and the total compensations for all of these sources. Additionally, the Udacity.com course designer created email features that give the total number of e-mails sent and received for each user, and the total number of emails sent to and receieved from a POI. Of the 144 people, 18 of them are labeled as POIs.

In regards to the financial information, I defined outliers as having values that are more than 3 standard deviations from the mean value for the group. This is not the traditional definition or criteria for an outlier which is 1.5 times the interquartile range below the first quartile or Above the third quartile. I used my definition after I have replaced missing values with zero. Using this definition of outliers there are 25 people in the data set that are financial outliers:

['ALLEN PHILLIP K','BELDEN TIMOTHY N', 'BHATNAGAR SANJAY', 'BLAKE JR. NORMAN P','FREVERT MARK A','GRAMM WENDY L','HANNON KEVIN P','HIRKO JOSEPH', 'HORTON STANLEY C', 'HUMPHREY GENE E', 'JAEDICKE ROBERT', 'LAVORATO JOHN J','LAY KENNETH L','LEMAISTRE CHARLES', 'MARTIN AMANDA K', 'MCCLELLAN GEORGE', 'PAI LOU L', 'RICE KENNETH D','SAVAGE FRANK','SHANKMAN JEFFREY A', 'SKILLING JEFFREY K', 'URQUHART JOHN A', 'WAKEHAM JOHN','WHITE JR THOMAS E', 'WINOKUR JR. HERBERT S']

This accounts for 17% of the data being considered an outlier and also contains 33% of the POI. Ultimately I decided that financial outliers were relevant information, and decided not to remove them from the data.

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”]

3. What algorithm did you end up using? What other one(s) did you try? [relevant rubric item: “pick an algorithm”]

I started off by creating two features I thought were be informative to the data: the ratio of emails received from a POI to the total number of emails received and the ratio of emails sent to a POI to the total number of emails sent.

I thought these features would be more informative than the total number of emails sent or received from a POI because it normalizes by how active or how popular a person is.

If a person sends 10 emails, and 5 of them are to a POI, that seems more relevant than if a person sends 1000 emails and 5 of them are to a POI. A person who sends 50% of their emails to a person of interest seems more suspect than a person who only sends 0.5% of their emails to a POI.

The inverse is not true, however. If a person received 10 emails from a POI, that is as relevant regardless if the person receives 20 emails or 2000 emails. The important idea is that how often is a POI is contacting this person, and how does that affect the likelihood that a person is also a POI. This ratio does not capture that idea, but I created it because I was willing to be proven wrong.

After I created these features I started with a systematic search of the results of pairs of features from the data set.

I trained two classifiers, a Decision Tree and a Gaussian Naive Bayes, on all pairs of features using 10 fold cross validation, and examined the recall and precision scores on the test set. At this point I did not tune any parameters, nor normalize the data. In fact both Decision Trees and Naive Bayes are not affected by normalization. If i chose an SVM, I would have normalized.

The best results of the scan is as follows: (These results vary slightly from run to run)

Type	Variable 1`	Variable 2	Recall	Precision
Tree	exercised_stock_options	from_this_person_to_poi_ratio	36.8%	38.9%
Tree	bonus	expenses	43.7%	38.8%
Tree	total_stock_value	other	36.8%	38.8%
NB	total_stock_value	deferred_income	53.8%	38.8%
Tree	expenses	other	47.4%	50.0%
Tree	expenses	from_this_person_to_poi	43.8%	38.8%

The decision tree consistently performed better than Naive Bayes, so I decided to to use Decision Trees for this analysis.

I next looked at my created features for the ratio of mails from or two POI.   I performed the same scan including one or two of my created features. The best results follow:

In conjunction with from_poi_to_this_person_ratio
(These results vary slightly from run to run)

Variable 1`	Variable 2	Importance	Recall	Precision
exercised_stock_options (0.43)	from_this_person_to_poi_ratio (0.42)	(0.14)	44.4%	44.4%
restricted_stock (0.21)	other (0.34)	(0.45)	47.1%	44.4%

In conjunction with from_this_person_to_poi_ratio
(These results vary slightly from run to run)

Variable 1`	Variable 2	Importance	Recall	Precision
total_payments (0.304)	bonus (0.369)	0.326	42.8%	50.0%
exercised_stock_options (0.249)	other (0.358)	0.392	52.9%	44.4%
total_stock_value (0.204)	other (0.350)	0.446	44.4%	44.4%

In conjunction with from_poi_to_this_person_ratio and from_this_person_to_poi_ratio
(These results vary slightly from run to run)

Variable 1`	Variable 2	Importance 1	Importance 2	Recall	Precision
to_messages (0.19)	exercised_stock_options (0.28)	0.42	0.14	43.4%	55.5%
total_payments (0.25)	exercised_stock_options (0.32)	0.31	0.11	42.1%	44.4%
exercised_stock_options (0.34)	restricted_stock_deferred (0.00)	0.42	0.24	44.4%	44.4%

exercised_stock_options (0.32)	loan_advances (0.00)	0.48	0.20	44.4%	44.4%
exercised_stock_options (0.38)	from_messages (0.10)	0.39	.14	47.3%	50.0%

In both cases using a single one of my created features in conjunction with two financial features performed better than using both when used with a decision tree. They did perform better than a decision tree with two variables. The importance of the to ratio tends to be low (unimportant), while the to ratio tends to be very important. This is consistent with the earlier description in the text.

I am curious how these values compare to using the three best features. That scan produces the following.
(These results vary slightly from run to run)

Variable 1`	Variable 2	Variable 3	Recall	Precision
expenses (0.31)	other (0.37)	from_this_person_to_poi (0.32)	66.7%	55.5%

This performed much better than using my features, so I opted to not use my features for the rest of the analysis.

In looking for the best features to use I scanned for 4 variables, importance are in parentheses:
(These results vary slightly from run to run)

Variable 1`	Variable 2	Variable 3	Variable 4	Recall	Precision
salary (0.13)	exercised_stock_options (0.30)	other (0.23)	from_this_person_to_poi_ratio (0.35)	58.8%	55.5%
to_messages (0.07)	expenses (0.25)	other (0.43)	from_this_person_to_poi (0.25)	55.5%	55.5%
deferral_payments (0.06)	expenses (0.30)	other (0.33)	from_this_person_to_poi (0.32)	62.5%	55.5%
deferral_payments (0.05)	expenses (0.44)	other (0.51)	director_fees (0.00)	52.6%	55.5%

exercised_stock_options (0.31)	total_stock_value (0.00)	other (0.29)	from_this_person_to_poi_ratio (0.40)	55.5%	55.5%
exercised_stock_options (0.21)	expenses (0.16)	other (0.34)	deferred_income (0.29)	62.5%	55.5%
restricted_stock_deferred (0.00)	expenses (0.36)	other (0.27)	from_this_person_to_poi (0.37)	70.8%	66.6%
total_stock_value (0.19)	expenses (0.15)	other (0.23)	from_this_person_to_poi_ratio (0.43)	55.5%	55.5%
expenses (0.25)	loan_advances (0.00)	other (0.43)	from_this_person_to_poi (0.32)	61.1%	61.1%
expenses (0.30)	from_messages (0.06)	other (0.33)	from_this_person_to_poi (0.31)	68.8%	61.1%
expenses (0.30)	other (0.39)	from_this_person_to_poi (0.32)	director_fees (0.00)	68.8%	61.1%

The results are not significantly improved by an additional feature, and with the exception of restricted_stock_deferred, the relative importance is 0. The best features to use for a decision tree seem to be ‘expense’, ‘other’ and ‘from_this_person_to_poi’. These are also the most repeated variables in the table results of 4 variables. One of my created variables does show up with high importance, but it does not give the best results in this scan.

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”]

Tuning an algorithm involving changing the parameters of a particular algorithm in the attempt to increase its performance, hopefully in a context broader than the training. If this is not done well, the algorithm can overfit the training data and perform significantly more poorly on test or real data. In the case of Decisions trees, the classification algorithms I used, this is a real danger because Decision trees tend to over fit.

After I decided on the features in the previous section, I used sklearn Grid Search to scan through the parameters to give the best performance on the training data using the following parameters over the listed values:

Split Criteria: Gini Index, Mutual Information/Entropy
Max Depths: None, 1, 2, 4, 8, 16, 32
Min Sample Split: 1, 2, 4, 8, 16, 32
Min Sample Leaves: 1, 2, 4, 16, 32

The search was performed over half of the total data set to avoid any possible overfitting. The best parameters found were ('gini',None,32,1). This classifier was passed to the classifier tester for final analysis.

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"]

Validation is an attempt to confirm that a model will give reasonable or consistent results on new, untrained data. A classic mistake is to test the results of a model on the data used to train the model. This is no doubt give the best possible score, but can over fit the data leading to less than desired results on new data. Validation protects against this mistake by training the model on one set of data and testing on yet another.

I used 10-Fold Cross Validation for investigating and comparing algorithms in this analysis. This is where there the algorithm is trained on the on the data 10 times using 90% of the data as a training set and 10% of the data as a testing set. Each time this is done, the training and testing set are shuffled to create a new 90/10 split on the data. I then used the average performance as an estimate of its performance on new data. It is through this method I settled on the Decision Tree algorithm over Naive Base, and selected the three features I choose for this analysis.

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"]

In making decisions in this analysis, I choose the metric Recall and Precision. Recall is the ratio of the number of POI's correctly identified out of the total number of POI. Precision is the number of POI's correctly identified out of all the values predicted to be POI's.

For my analysis I found that my average recall was 36.8%, and my average precision was 48.0%.

In terms of the data, the recall means that, on average, it correctly identifies 6.5 out of the 18 POI in the data set.   The precision means about half of all the people it predicts are POI, turn out to actually be POI.

Note: The code review shows a critical error that I can not reproduce on two different machines. I surrounded the place in question with a try/except where I redefine the classifier with the classifier round on my machine.