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”]

Overall data and background

The goal of this project is to built a person of interest identifier based on the financial data and the enron email provided. There are 146 data points which translates to persons in the Enron dataset. Each person is consist of 21 features. Out of 146 person, there are 18 person which has a feature called POI (Person of interest). We know that the data is incomplete as there should be 35 existing POIs but we only have 18 in our dataset, but that does not stop us from mining the data and build a classifier on it.

Outliers

The outliers are basically, titles such as “The Travel Agency in the park” or “Total” which are not related to the POI. Those data points are removed immediately when they were found. Besides that going through the data points I also found out that LOCKHART EUGENE E’s data point is consist of NaN and 0, I removed this person from the dataset as well. So in the end, we ended up with 143 data points for our algorithm.

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”]

I started out with all finance features, email features and my newly created features when doing my feature selections using Extra Tree Classifier’s feature importance. That’s a total of 21 features. I removed the low ranking features one by one to observe the changes of precision and recall. The scores peaks at about 10 features and then it gradually drops and become stable at around 8 ~ 5 features. The peak/highest precision and recall score which I get is around 3 features and then it starts to drop dramatically when it is 2 ~ 1 features. We wanted our classifier to correctly identify a real POI when it shows up therefore the recall score is important for us and also we want to have the minimum number of features so that our classifier does not overfit and capture the maximum variance. So as a result these 3 features were chosen: *bonus, total_stock_value, exercised_stock_options*

Below are my final features:

bonus	0.365936828
total_stock_value	0.36685546
exercised_stock_options	0.267207712

	Features	Accuracy	Precision	Recall	F1
21	fraction_to_poi, bonus, exercised_stock_options, total_stock_value, expenses, total_payments, deferred_income, other, restricted_stock, from_poi_to_this_person, long_term_incentive, fraction_from_poi, salary, shared_receipt_with_poi, from_this_person_to_poi, to_messages, from_messages, deferral_payments, restricted_stock_deferred, loan_advances, director_fees	0.86113	0.44147	0.1565	0.23108
20	exercised_stock_options, fraction_to_poi, other, salary, restricted_stock, bonus, long_term_incentive, shared_receipt_with_poi, deferred_income, total_payments, expenses, total_stock_value, fraction_from_poi, from_this_person_to_poi, to_messages, from_poi_to_this_person, deferral_payments, from_messages, loan_advances, restricted_stock_deferred	0.86427	0.47414	0.165	0.24481
19	exercised_stock_options, fraction_to_poi, bonus, salary, deferred_income, total_stock_value, other, shared_receipt_with_poi, from_this_person_to_poi, restricted_stock, from_messages, total_payments, long_term_incentive, from_poi_to_this_person, expenses, to_messages, fraction_from_poi, deferral_payments, loan_advances	0.86307	0.4624	0.166	0.2443
18	total_stock_value, deferred_income, fraction_to_poi, exercised_stock_options, shared_receipt_with_poi, restricted_stock, bonus, expenses, total_payments, other, long_term_incentive, from_messages, salary, from_poi_to_this_person, fraction_from_poi,	0.86133	0.44505	0.162	0.23754

	from_this_person_to_poi, deferral_payments, to_messages				
17	exercised_stock_options, bonus, total_stock_value, other, deferred_income, expenses, fraction_to_poi, restricted_stock, fraction_from_poi, from_this_person_to_poi, from_poi_to_this_person, shared_receipt_with_poi, salary, long_term_incentive, from_messages, deferral_payments, total_payments	0.8634	0.46658	0.171	0.25027
16	total_stock_value, exercised_stock_options, fraction_to_poi, bonus, expenses, shared_receipt_with_poi, fraction_from_poi, from_this_person_to_poi, deferred_income, other, salary, long_term_incentive, from_poi_to_this_person, restricted_stock, from_messages, deferral_payments	0.86107	0.44802	0.181	0.25783
15	shared_receipt_with_poi, deferred_income, exercised_stock_options, long_term_incentive, expenses, total_stock_value, salary, bonus, other, restricted_stock, fraction_to_poi, from_this_person_to_poi, fraction_from_poi, from_messages, from_poi_to_this_person	0.86313	0.46241	0.163	0.24104
14	fraction_to_poi, total_stock_value, bonus, deferred_income, other, exercised_stock_options, expenses, fraction_from_poi, salary, shared_receipt_with_poi, from_messages, restricted_stock, from_this_person_to_poi, long_term_incentive	0.86333	0.46334	0.158	0.23565
13	exercised_stock_options, salary, bonus, deferred_income, fraction_to_poi, restricted_stock, expenses, shared_receipt_with_poi, from_this_person_to_poi, total_stock_value, other, fraction_from_poi, from_messages	0.86247	0.45619	0.164	0.24127
12	fraction_to_poi, total_stock_value, deferred_income, bonus, exercised_stock_options, other, shared_receipt_with_poi, salary, expenses, fraction_from_poi, restricted_stock, from_this_person_to_poi	0.86287	0.46101	0.1685	0.2468
11	expenses, deferred_income, fraction_to_poi, exercised_stock_options, total_stock_value, other, bonus, shared_receipt_with_poi, fraction_from_poi, salary, restricted_stock	0.85933	0.43309	0.178	0.2523

10	total_stock_value, other, salary, fraction_to_poi, expenses, bonus, exercised_stock_options, shared_receipt_with_poi, deferred_income, fraction_from_poi	0.86627	0.4965	0.213	0.29811
9	bonus, total_stock_value, exercised_stock_options, shared_receipt_with_poi, deferred_income, expenses, other, fraction_from_poi, salary	0.86073	0.44949	0.198	0.2749
8	exercised_stock_options, deferred_income, bonus, expenses, total_stock_value, shared_receipt_with_poi, fraction_from_poi, other	0.869	0.523	0.199	0.2883
7	shared_receipt_with_poi, bonus, deferred_income, total_stock_value, expenses, exercised_stock_options, fraction_from_poi	0.85914	0.51707	0.212	0.30071
6	exercised_stock_options, total_stock_value, bonus, expenses, deferred_income, shared_receipt_with_poi	0.85886	0.51395	0.221	0.30909
5	total_stock_value, bonus, exercised_stock_options, deferred_income, expenses	0.84864	0.43972	0.217	0.29059
4	exercised_stock_options, bonus, total_stock_value, deferred_income	0.85286	0.46951	0.231	0.30965
3	bonus, total_stock_value, exercised_stock_options	0.86015	0.58585	0.3105	0.40588
2	total_stock_value, bonus	0.83338	0.44046	0.307	0.36181
1	total_stock_value	0.77154	0.23925	0.2225	0.23057

New Features

I created two new features which are fraction_from_poi and fraction_to_poi, these two features are ratios that represent the number of emails sent to/from a POI over the total email sent/received.

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

Algorithm

The final algorithm which I used is Extra Tree Classifier which is basically a cheaper algorithm to train when compared to Random Forest Classifier. It does not require any feature scaling as it basically partitions the data into 2 sets by comparing the features to a threshold value.

I tried out 4 different types of algorithm with my final feature selection. First was Naive Bayes, I got a high recall and low precision and there was no parameter settings tuning. I then try Decision Tree Classifier, the recall improved but precision drop. I then try random forest as I thought it would improved further on the results given the case when Random Forest do sample splitting, samples are drawn from a bootstrap of sample instead of the whole and splits are chosen completely from a random subset of features. I proceed further with parameter tuning but was not manage to get a better recall score. Lastly I resorted to Extra Trees Classifier. It gives me a higher precision and recall score compare to the rest of the algorithms.

Naïve bayes	Precision: 0.48581	Recall: 0.35100
Decision Tree	Precision: 0.36321	Recall: 0.38300
Random Forest	Precision: 0.59147	Recall: 0.29600
ExtraTrees	Precision: 0.57115	Recall: 0.314

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"]

Algorithm Tuning

By tuning the parameters of an algorithm, we can find a better fit for our data points and get the best performance out of it, if we don't do this well we may under/over fit the data against our algorithm.

I used GridSearchCV to automatically tune the best parameters setting for the algorithm I choose. The parameters which I used in GridSearchCV were param_grid which tune the parameters settings which I specify. GridSearchCV uses accuracy by default as the metric to decide which parameter setting is best, I change the scoring='recall' as our goal for this project is to try to get the precision and recall to be higher than .3. I also passed in the cross validation generator setting as StratifiedShuffleSplit to allow training and testing on all data available.

For Extra Tree Classifier the main parameter to adjust is **n_estimator**, **max_features, criterion**. From SKlearn's documentation it stated that good result can be achieved when

max_depth=None and min_samples_split=1. But I set it to see and adjustment needs to be made.

parameters =
{'n_estimators':[1,5,10,15,20],'criterion':['gini','entropy'],'max_features':['sqrt','log2',None],
'min_samples_split':[1,2,4,6,8,10], 'max_depth': [None, 4, 10, 15]}

cross_validation=StratifiedShuffleSplit(labels, n_iter=1000, test_size=.1, random_state = 42)
clf = GridSearchCV(clf, parameters, cv=cross_validation, scoring = 'recall')

The final parameter settings which I got is
min_samples_split=1, n_estimators=1, max_features=None, criterion='gini', max_depth='None'
The best recall score I get is **0.4105**

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

Validation is a technique used to test our model on an independent dataset (Test Data) against a another dataset (Train Data) to get an estimate of performance. It also serves as a check on overfitting. If we made a mistake, we might end up with a model that will perform well on our training data but not on other data set.

StratifiedShuffleSplit is suitable for our problem because it randomized the data while retaining the data in a balanced manner because if it is not randomized it is possible to split the training and testing set in a way that we might have a training set that contains no POIs while the test set has real POIs in it.

When I finalized my feature selection. I used StratifiedShuffleSplit and GridSearchCV to cross validate the precision and recall score. The result remains the same before proceeding with algorithm selection

For comparing algorithm performance and algorithm selection, parameters options were provided to GridSearchCV and StratifiedShuffleSplit since we want to have a balanced dataset used for cross validation.

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"]

Results

There are three metrics which I used to evaluate how my algorithm perform. They are **accuracy**, **precision** and **recall.** Below is the result which I obtained after running my algorithm several times:
Accuracy: 0.81654
Precision: 0.40559
Recall: 0.4065
Accuracy means how well the algorithm predicted the person is or isn't a poi from the dataset.
Precision means how confident are we in identifying that the person is a real poi.
Recall means the algorithm is able to identify a POI every time when its show up in the dataset.