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”]

<u>Revised for 3rd submission</u>

I continue to remove low ranking features to observe the changes of the accuracy, precision and recall. I noticed that the precision and recall starts to drop as I continue to remove more features. It is noticeable that the score remains stable at around 8 to 6 features. However, there’s a significant change in recall between 1 or 2 features. It seems to produce the highest recall score I recorded so far but the accuracy drop a fair bit. Therefore, this leads me to stick with my top 10 features which I stated previously.

Features	Accuracy	Precision	Recall
9	0.85807	0.43087	0.201
8	0.85813	0.42	0.168
7	0.85827	0.42065	0.167
6	0.84614	0.40811	0.171
5	0.83893	0.35053	0.1495
4	0.83964	0.37306	0.18
3	0.84943	0.43182	0.171
2	0.83069	0.4268	0.293
1	0.77077	0.23741	0.2215

Revised for 2nd submission

I included all finance and email features for my initial feature selection except email address feature which is a text string. I used Extra Tree Classifier to select my feature importance. First attempt, the features which I removed was either giving me a importance less than 0.05 or falls out of my top 10 feature list. I did my feature importance selections again without new engineered features, the precision and recall seems to drop. I then added 2 new engineered features plus my top 10 feature list and rerun again, the impact on precision and recall was not that significant. I then remove some features until I get a final 10 top feature list, the precision and recall seems to achieved the highest score.

	Accuracy	Precision	Recall
All Features	0.86047	0.43309	0.1505
Top 10 Features without Engineered Features	0.85193	0.3683	0.1545
Top 10 Features with 2 Engineered Features	0.85753	0.40999	0.156
Final Top 10 Features	0.86353	0.47251	0.202

Below are my final features:

fraction_to_poi	0.1220246031
exercised_stock_options	0.1161553027
deferred_income	0.1060981121
total_stock_value	0.1027474608
expenses	0.08800671084

other	0.08527566813
bonus	0.08095121584
from_poi_to_this_person	0.07355696377
total_payments	0.06443340588
salary	0.06155869507

Feature selection

I started out by using all the finance features. I reached these final features by using Extra Tree Classifier to identify the least important feature and eliminate them one by one. The features which I removed was either giving me a importance less than 0.05 or falls out of my top 10 feature selections.

Below are my final features:

exercised_stock_options	0.113505198
other	0.111392683
expenses	0.09354477
fraction_to_poi	0.093169172
total_stock_value	0.092851773
bonus	0.083783854
salary	0.081875736
deferred_income	0.080555792
long_term_incentive	0.074562272
fraction_from_poi	0.060690868

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 compare 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 low recall and there was no parameter settings tuning. I then try Decision Tree Classifier, the recall improved a bit but not a lot, precision remains almost the same whereas accuracy drop a bit. 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 recall of .3 or better. Lastly I resorted to Extra Trees Classifier. It gives me a higher recall than the rest of the algorithm.

Naïve bayes	Accuracy: 0.86000	Precision: 0.43797	Recall: 0.17650
Decision Tree	Accuracy: 0.85967	Precision: 0.43757	Recall: 0.18400
Random Forest	Accuracy: 0.86360	Precision: 0.46982	Recall: 0.17900
ExtraTrees	Accuracy: 0.86080	Precision: 0.44919	Recall: 0.19450

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.

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=100, random_state = 50)
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='15'
The best recall score I get is **0.395**

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.

When doing GridSearchCV, I passed in a cross validation generator which is StratifiedShuffleSplit to ensure that we get the best parameters tune and to validate that my algorithm is correct.

Revised for 3rd submission
I used cross_val_score to evaluate which feature selection and algorithm performs the best. I set the K-fold to be 10 to allow it to compute score 10 consecutive times, scoring parameter is set to get the best recall score and final step would be to get the mean of all 10 scores. The final algorithm and feature is selected based on the highest score. The final features list was selected based on systematically removing the lowest features importance one at a time until I

am satisfied with the cross validation accuracy, precision and recall. The cross validation was done using startified shuffle split cross validation provided in the starter code.

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"]

Revised for 2nd submission
Accuracy: 0.82640
Precision: 0.35701
Recall: 0.37700

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: 0.82153
Precision: 0.34190
Recall: 0.36600
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.