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”]

Goal of this project is to predict persons of interest for the Enron scandal using Machine learning techniques. The given dataset has both financial and e-mail information of various people connected to Enron. Initial analysis of the data using Pandas and visual inspection shows few outliers. One is Total and the other one is THE TRAVEL AGENCY IN THE PARK. Since we are only focused on Persons of Interest, these two outliers should be removed. Also LOCKHART EUGENE E, who has no non NaN values was removed as well.

I removed these three outliers from the dictionary. Please see below overall characteristics of the data.

Count of POIs in the dataset 18

Count of non-POIs in the dataset 128

No. of records with missing total_payments 21

No. of records with missing salary 51

No. of records with missing bonus 64

No. of records with missing director_fees 129

No. of records with missing restricted_stock_deferred 128

No. of records with missing deferral_payments 107

No. of records with missing from_messages 60

No. of records with missing from_poi_to_this_person 60

No. of records with missing from_this_person_to_poi 60

No. of records with missing shared_receipt_with_poi 60

No. of records with missing to_messages 60

No. of records with missing deferred_income 97

No. of records with missing exercised_stock_options 44

No. of records with missing expenses 51

No. of records with missing loan_advances 142

No. of records with missing long_term_incentive 80

No. of records with missing restricted_stock 36

No. of records with missing total_stock_value 20

No. of records with missing other 53

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”, “intelligently select features”, “properly scale features”]

I created the following two new features and used them in the initial analysis -

ratio_of_messages_received_from_poi – which is a ratio of number of messages received from poi to total no of messages received by the individual

ratio_of_messages_sent_to_poi - which is a ratio of a number of messages sent to poi to total number of messages sent out by the individual.

I came up with these features with the thought that the ratio of communications with POIs to the overall communication a person does might be valuable for this analysis. My hunch proved correct and ration_of_meesges_sent_to_poi was selected as one of the important feature by kbest.

1. For the initial analysis I used 11 features including the two I created. Also I have used feature scaling to give equal weightage to both email features and financial features Below table shows the importance and accuracy scores.

Feature	**Importance**
exercised_stock_options	0.255473492355
Shared_receipt_with_poi	0.226004511278
expenses	0.180751044277
ratio_of_messages_sent_to_poi	0.135602729245
total_payments	0.115005291005
bonus	0.0423703703704
restricted_stock	0.0281470588235
total_stock_value	0.0166455026455
ratio_of_messages_received_from_poi	0.0
salary	0.0
long_term_incentive	0.0

Accuracy Scores using above features

Accuracy	0.825
Precision	0.327
Recall	0.291
F1	0.308

2. In the next step I utilized skbest, univariate feature selection process, with feature scaling. I gave skbest a parameter option to select 5 features. Below table shows the features selected by skbest and the importance scores.

Feature	**Importance**
exercised_stock_options	0.319567174272
ratio_of_messages_sent_to_poi	0.235173099616
bonus	0.231792344888
salary	0.136598477668
total_stock_value	0.076868903557

Accuracy Scores using the above features

Accuracy	0.801
Precision	0.287
Recall	0.333
F1	0.308

Selecting 5 best features using skbest had a slight negative impact on the accuracy scores, but not significant.

3. In an effort to see if I can further reduce the number of features, I utilized skbest again, with feature scaling. I gave skbest a parameter option to select only 3 features. Below table shows the features selected by skbest and their importance.

Feature	**Importance**
total_stock_value	0.353209053981
exercised_stock_options	0.348411988654
bonus	0.298378957365

Accuracy Scores using the above features

Accuracy	0.803
Precision	0.285
Recall	0.318
F1	0.301

Table below summarizes various options.

No. of features used for analysis	Accuracy	Precision	Recall	F1
11 features	0.825	0.327	0.291	0.308
5 features	0.801	0.287	0.333	0.308
3 features	0.803	0.285	0.318	0.301

In an effort to maximize precision/recall scores and at the same time keep the no. of features to a minimum, I chose to go with 5 features using skbest selection process. The following features were selected.

Features
exercised_stock_options
ratio_of_messages_sent_to_poi
bonus
salary
total_stock_value

3. 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”]

   I ended up using Decision Tree. Both Naïve Bayes and SVM also had high accuracy in the initial analysis, when trained and tested using Train/Test split cross validation method. I explored Decision Tree further with feature scaling, feature selection and parameter tuning. Decision Tree gave better result at the end and I chose to go with Decision Tree as the final classifier.

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? What parameters did you tune? (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 items: “discuss parameter tuning”, “tune the algorithm”]

   Parameter turning means optimizing the parameters used in the classifier for the given data to improve prediction accuracy. I have used GridSearchCV function to tune the parameters for the Decision Tree classifier. If the parameters are not tuned properly, it will impact accuracy, precision and recall scores.The following parameters were tuned

```
parameters = { 'DTC__criterion': ['gini', 'entropy'],

        'DTC__min_samples_split': [2, 10, 20],

        'DTC__max_depth': [None, 2, 5, 10],

        'DTC__min_samples_leaf': [1, 5, 10],
```

'DTC__max_leaf_nodes': [None, 5, 10, 20]

Below table shows the scores before and after tuning the parameters using Decision Tree classifier.

	Accuracy	**Precision**	**Recall**	**F1**
Before tuning	0.801	0.287	0.333	0.308
After tuning	0.874	0.533	0.476	0.502

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 items: "discuss validation", "validation strategy"]

Validation is a way of verifying the accuracy of your trained classifier on independent data. One of the classic mistakes is to overfit your data. It is crucial to avoid overfitting and making sure that our classifying algorithm is usable in practical datasets. In case validation is not done properly, the resulting classifier algorithm is very likely to perform poorly on practical datasets, other than the one used in training and evaluating phases.

In the initial analysis I have used train/test split method for validation. Since we have a very limited data, Train/Test is not most effective. For the final chosen classifier, I have used StratifiedShuffleSplit method for validation. The dataset is small and skewed towards non-POI, we need a technique that accounts for that or the risk is that we would not be able to assess, in the validation phase, the real potential of our algorithm in terms of performance metrics. The chance of randomly splitting skewed and non representative validation sub-sets could be high, therefore the need to use stratification (preservation of the percentage of samples for each class) to achieve robustness in a dataset with the aforementioned limitations..

6. Give at least two 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"]

The following evaluation metrics were used:

Precision score for a class is the *number of true positives divided by the total number of elements labeled as belonging to the positive class*). In the project context, it is the ratio of number of true POIs correctly identified to the total number of records identified as POIs

Recall score Recall in this context is defined as the *number of true positives divided by the total number of elements that actually belong to the positive class* (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been). In the project context, it is the ratio of no. of true POIs correctly identified to the total number of POIs in the dataset.