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”]

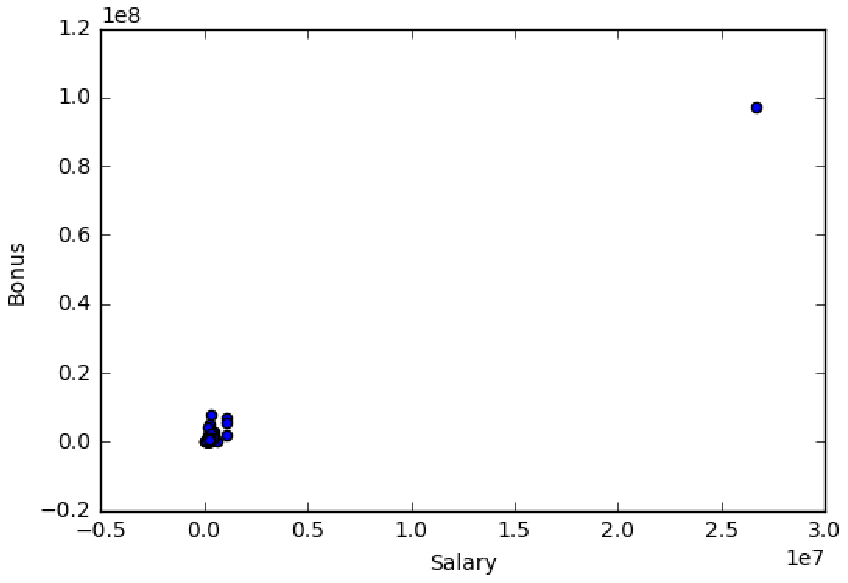
The aim of the project is to identify the persons involved in the infamous enron scandal.A person of interest(POI) is referred as someone involved in enron case,was indicted for fraud,settled with government,or testified in exchange for immunity.Their schemes included selling assets to shell companies at the end of each month and buying them back at the beginning of next month to avoid accounting losses.Also,they were responsible for causing electrical grid failures in California.

Here,we cannot easily assign the class label to the persons(POI or not a POI) manually by just looking at the data.However,using machine learning techniques,we can pick out those email and finance features which are more powerful in identification of the POIs.For eg.maybe POIs more frequently contacted other POIs through emails and so on.

There are 146 persons in the enron dataset which include 18 POIs,when actually there are 35.We have 21 features for each person.

The POI labels do not have any missing values which is good,otherwise the accuracy of the predictions will be affected.But there are lot of missing values in the columns like loan_advances,so there is no use in including it among the features.

An outlier detected among the data points is the TOTAL,which is a spreadsheet quirk.Since it is not a real person,it is removed manually.



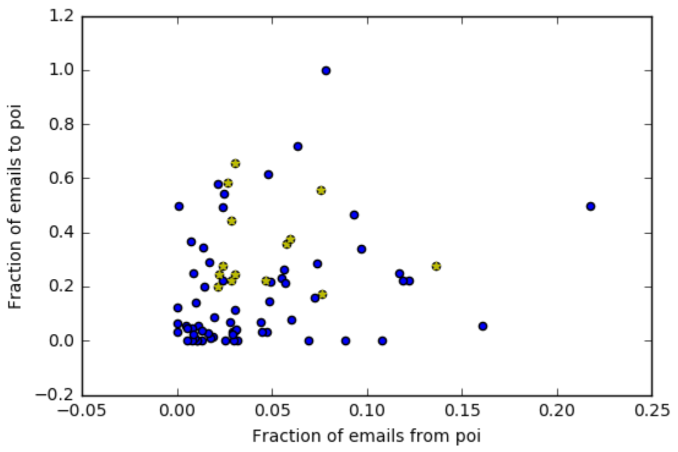
After removing it,two more outliers pop up(LAY KENNETH and SKILLING JEFFREY).But these are valid data points so we keep them.

Feature	Number of non-missing values
salary	95
to_messages	86
deferral_payments	39
total_payments	125
exercised_stock_options	102
bonus	82
restricted_stock	110
shared_receipt_with_poi	86
restricted_stock_deferred	18
total_stock_value	126
expenses	95
loan_advances	4
from_messages	86
other	93
from_this_person_to_poi	86
poi	146
director_fees	17
deferred_income	49
long_term_incentive	66
email_address	111
from_poi_to_this_person	86

We can see that the feature we need to predict i.e poi does not have any missing values.But there are features like loan_advances that have a lot of missing values and hence it would not be beneficial in increasing the accuracy of prediction model.

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

A set of 5 features were chosen for feeding into the machine learning algorithm.
Under the assumption that POIs contacted other POIs more frequently,I checked if there is any relation between the number of emails from this person to POI('from_this_person_to_poi') and number of emails from POI to this person('from_poi_to_this_person').But there was not any strong pattern,so I wondered if the fraction of emails that come to a person from POI or the fraction of emails that a person sends to POI mattered.
So,two new features were created:’fraction_to_POI’ and ‘fraction_from_POI’ .
fraction_from_POI:the fraction of emails that a person receives that comes from a POI which is obtained by dividing the from_POI_to_this_person by from_messages
fraction_to_POI:the fraction of emails that a person sends to POI which is obtained by dividing the from_this_person_to_POI by to_messages.



We can see that it's a very powerful feature.If less than 20% of the messages you send are sent to POI,it is almost certain that you are not a POI yourself.
After creating the new features and adding it to the features list,I used **SelectKbest** to find out the most powerful features.A set of 5 most powerful features were chosen by analysing the scores.It is shown in the table below.

Feature selected	Score from SelectKbest
salary	18.5757
fraction_to_POI	16.641
Bonus	21.06
total_stock_value	24.467
exercised_stock_options	25.095

After selecting the features,I did a feature scaling using MinMaxScaler.This is because the Enron data consists of both financial features(salary,bonus,etc) and email features have magnitudes of varying ranges and hence they should be rescaled to comparable ranges.This is especially important when using algorithms like k nearest neighbor where distance calculation of the data points to the cluster centre is involved.

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”]

After trying around 4 algorithms,I settled with decision tree classifier and k nearest neighbor classifier algorithms.Without any tuning,they give pretty good performance in terms of accuracy,precision and recall scores.
Before tuning,

Algorithm	Accuracy	Precision	Recall
Decision tree classifier	0.78921	0.29227	0.3450
K Nearest Neighbor	0.87493	0.64164	0.282

The accuracy and precision score seemed to be higher for K Nearest neighbor.So,I picked that algorithm.However,I tuned the parameters for both the algorithms to see if there was an improvement in the metrics as discussed in the next section.

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 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”]

Tuning an algorithm is important since all of the estimator method and its parameters could be vary depend on the problem that we have. By tuning the algorithm, we will fit the parameters to our specific problem.If we don’t tune the parameters well,we won’t get the best performance.
For the decision tree classifier I used,the beat parameter values were obtained by using GridSearchCV.Range of values for min_samples_leaf and min_samples_split was passed and the results were:

min_samples_split=4,min_samples_leaf=4
But the parameters obtained by this method did not improve the precision and recall.The results after manually changing the parameters are shown below.
Decision tree

min_samples_split	min_samples_leaf	Accuracy	Precision	Recall
4	4	0.81429	0.27477	0.183
3	1	0.791	0.26873	0.269
5	1	0.79614	0.28369	0.28

For K Nearest Neighbor classifier,I manually changed the number of neighbor parameter,n_neighbors.The results are shown below for various values.

n_neighbors	Accuracy	Precision	Recall
3	0.87429	0.61928	0.31150
4	0.86979	0.67319	0.1720
5	0.87493	0.64164	0.282

Thus,**K Nearest neighbor** algorithm was chosen with n_neighbors=3.

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 involves splitting your data into testing and training set so that after fitting your classifier on the training data,we can check how well the algorithm performs by making predictions using the testing data.But the classic mistake we can make by just splitting the data into say 80:20 ratio is a problem because as in the enron data set,we can see there are only 18 POIs out of 146 data points,so there maybe a case where the training data does not contain a case of POI at all i.e the data is highly skewed.So,to make our algorithm learn better,I used K Fold cross validation where you subset the data into k bins of equal size and pick one those subsets as testing set ,the remaining (k-1) bins as training set.It is run multiple times,each time with a different bin as testing set.Here,the assessment of the learning algorithm will be more accurate.

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"]

The two evaluation metrics I used were precision and recall.I also used accuracy but it is not ideal for skewed classes.For the final tuned algorithm,
Precision=0.61928
Precision is calculated as True positives / (True positives+False positives).In our context,the proportion of correct prediction of people who are supposed to be poi.
Recall=0.31150
Recall is calculated as True positives/(True positives+False negatives).It means the proportion of poi,the model can detect of all poi.
For fraud prediction models like these,higher recall rate is preferred even if accuracy is sacrificed.
The identifier doesn't have a really high recall rate,but it does have good precision.That means that whenever a POI gets flagged in my dataset,I know with a lot of confidence that it is very likely to be a real POI and not a false alarm.On the other hand,the price I pay for this is that sometimes I miss real POIs since I'm effectively reluctant to pull trigger on edge cases.