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?

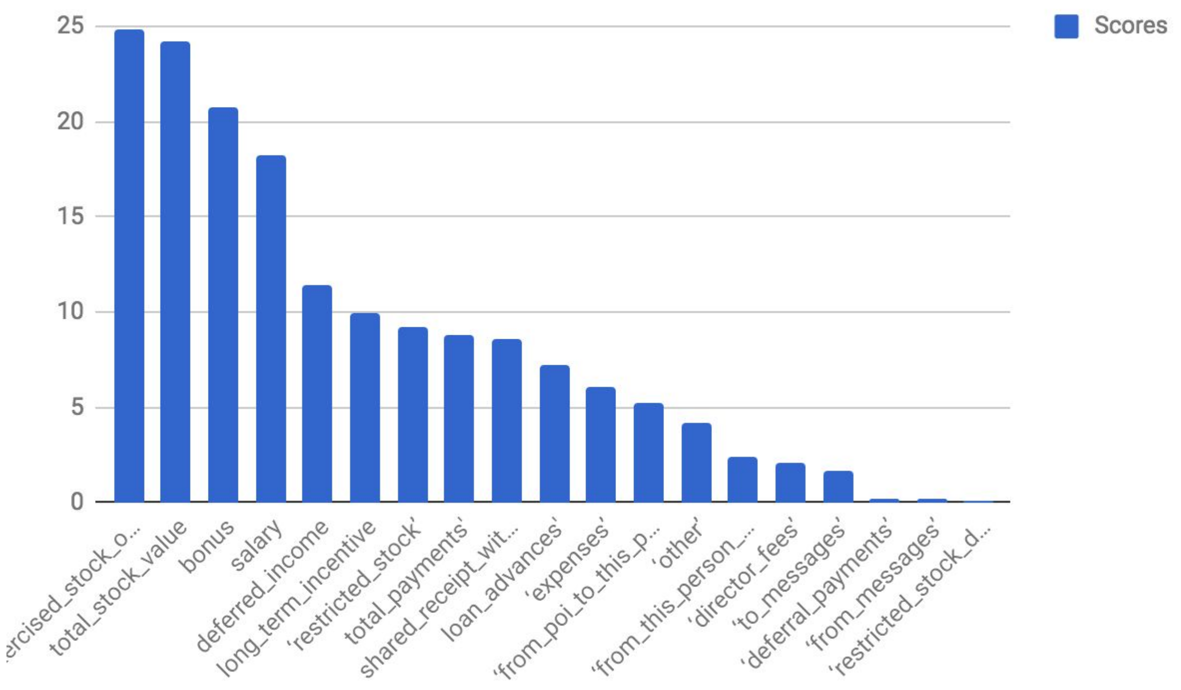
The purpose of this project is to try to building an algorithm to identify Enron employees who might have got involved with Enron scandal. The dataset has 146 observations, including 18 POIs and 128 non-POIs. Moreover, there are 14 financial features and 6 email features. The biggest outlier I can find so far is ‘TOTAL’, the second outlier I decided to delete is 'LOCKHART EUGENE E', since all its 14 financial features are ‘NaN’. The last outlier I decide to get rid of is ‘THE TRAVEL AGENCY IN THE PARK’, because it is an agency not a person in Enron. I decided to clean it out in order to avoid negative affect.
The number of missing values for each feature:

salary	51
to_messages	60
deferral_payments	107
total_payments	21
loan_advances	142
bonus	64
email_address	35
restricted_stock_deferred	128
total_stock_value	20
shared_receipt_with_poi	60
long_term_incentive	80
exercised_stock_options	44
from_messages	60
other	53
from_poi_to_this_person	60
from_this_person_to_poi	60
deferred_income	97

expenses	51
restricted_stock	36
director_fees	129

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"]

Here is a graph of features from highest score to lowest score.



As we can see from the graph above, the score of features ranged widely. I used minmax scaling to linearly rescale each feature to a common range. The features I end up using are the best 4 features which are scored above 18. They reason to use the four best features is that there is a large drop between the fourth and fifth fatures. They are: 'exercised_stock_options', 'total_stock_value', 'bonus', 'salary', I then took sklearn's MinMaxScaler to scale features which affects the performance of the tested algorithm.

I make two new features ‘msg_to_poi_ratio’ and ‘msg_to_poi_ratio’. The intuition is that I think the Enron scandal was not widely knew in the company, so only a small group of people knew it. Therefore, the number of emails sent among the group of people should be large, along with high ‘msg_to_poi_ratio’ and high ‘msg_to_poi_ratio’. However, when I operated my algorithm with new features, it turned out that the performance was worse than the algorithm without new features.

	With New Features	Without New Features
Accuracy	0.85	0.84
Precision	0.45	0.41
Recall	0.28	0.31

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 tried three algorithms Naive bayes, K-means, and svm. Naive bayes performed the best, followed are K-means and svm.

Algorithm	Accuracy	Precision	Recall
Naive bayes	0.85	0.45	0.28
K-means	0.68	0.45	0.33
SVM	0.85	0.22	0.45

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"]

Machine learning is a technique helps human to build model from data automatically. However, in order to get good results and performance, you need to set parameters for algorithm you want to operate. Tuning an algorithm is a process to figure out parameters which can optimize the the model in order to enable the algorithm to perform the best performance.

I built 3 algorithms, and used GridSearchCV function to get the best parameters for each of them:

- naive bayes: the model is simple and there's no need to specify any parameter
- kmeans: n_clusters is set to 2; tol (relative tolerance) is set to 1, and random_state (use to reproduce and confirm the result by fixing the seed) is set to 42
- svm: kernel is 'linear'; C is 1; gamma is 1; random_state is 42

5:What is validation, and what’s a classic mistake you can make if you do it wrong? How did you validate your analysis?

Validation is a technique to determine how well our model trained and to estimate model properties. A classic mistake I make is overfitting. In this case, model can perform pretty well, while do badly job on new or unseen data. In order to overcome it, I tried to use as less parameters as possible. On top of that, this Enron data is an imbalanced dataset. I then defined a function called evaluate_clf in which I applied cross validation technique to split the data into training data and test data 100 times, calculate the accuracy, precision, and recall of each iteration; then I took the mean of each metric.

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”]

I took three evaluation metrics: accuracy, precision and recall. Accuracy represent the ratio of no. of data points labeled correctly to no. of total data point. Precision measures true positive, meaning out of items labeled as positive how many truly belong to the positive class. 0.43 indicates that total 100 persons classified as POIs, 43 persons are actually POIs. Recall indicates a algorithm’s ability to classify positive among all items that are true positive. 0.37 means that given 100 true POIs in the dataset, this algorithm can classify 37 POIs.

Metric	Performance on Test Set	Performance on Final Set
Accuracy	0.854761904762	0.85
Precision	0.432977633478	0.5
Recall	0.373191558442	0.32