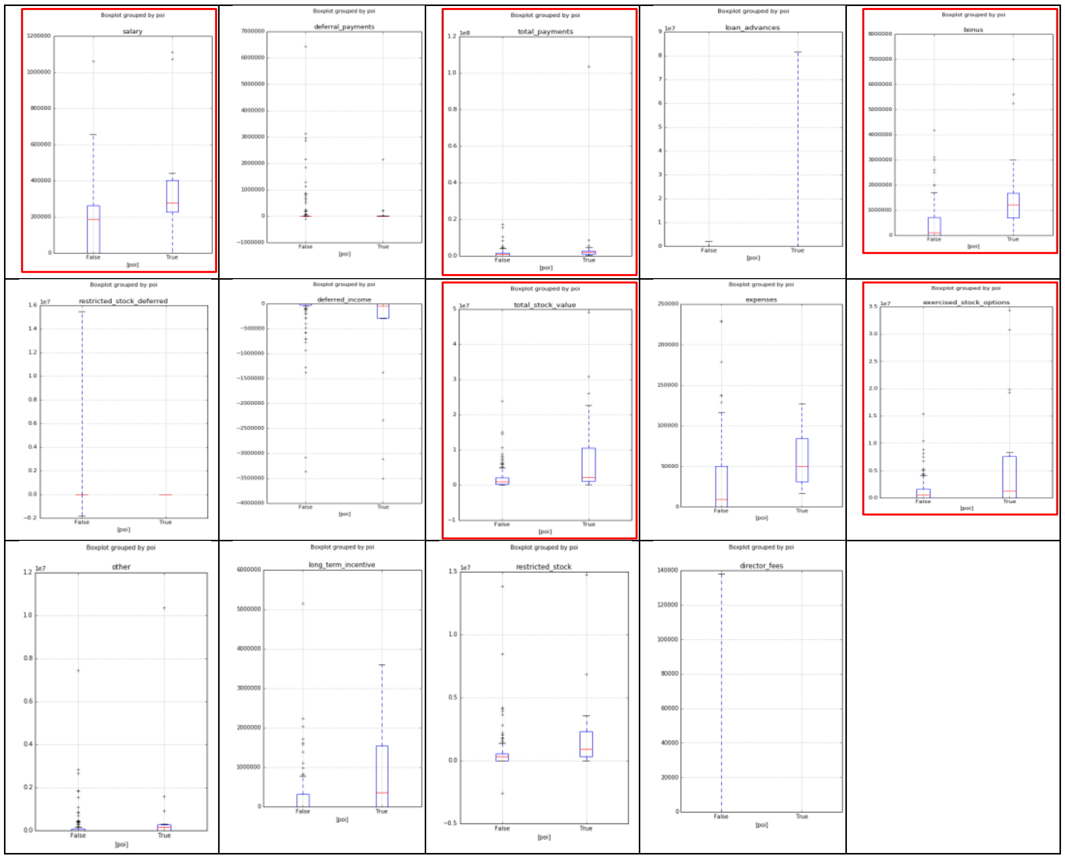
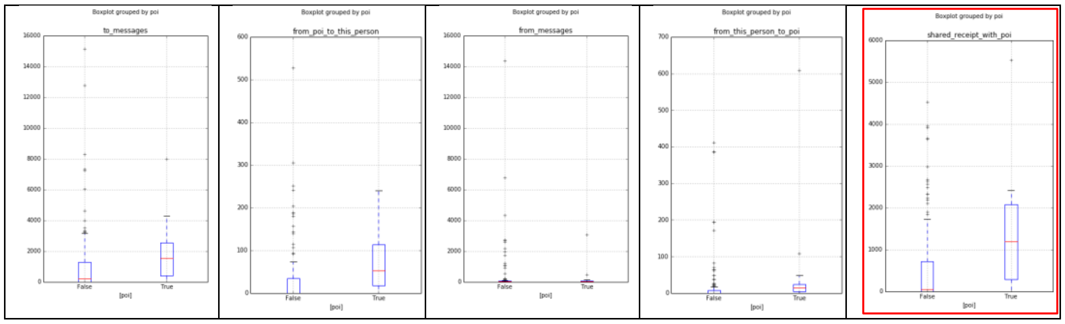
Project 5 Enron Submission Free-Response Questions

1. The goal of this project is to build a model and predict fraud. The data used to build model includes financial data and emails data. Using machine learning especially text learning can transform emails into numbers and identify frauds together with financial data.
   There are 146 suspects and 20 features for each suspect and one label column for if a POI. Some financial features contain large amount of NA values. And I detected there is one outlier with the name of 'Total'. This is not a suspect, so I deleted it from the data.

2. I picked those features as POI identifier, 'total_stock_value', 'bonus', 'salary', 'exercised_stock_options', 'total_payments', 'fraction_from_poi', 'fraction_to_poi' and 'shared_receipt_with_poi'. Since I use decision tree as the algorithm, I used exploratory data analysis and plot the distribution as box chart, to find the features having most different distribution and statistics between POI and non-POI as identifiers. The box charts showing that following features having a major difference between poi and non-poi considering both quantiles and range: salary, total_payments, bonus, total_stock_value, exercised_stock_options, shared_receipt_with_poi (colored in red in below box charts)





I also used SelectKBest, and order the features based on the results. The best three feautres are ‘bonus', 'salary', and 'fraction_to_poi', which match in my previous finding. My strategy is that to try the features I found having the most differences between poi and non-poi and those new features created, and if it doesn’t work well, I will try more top features from SelectKBest. The result turns out to work well so I didn’t try those other top features from SelectKBest.
Ranked features from high to low:
['bonus', 'salary', 'fraction_to_poi', 'deferred_income', 'long_term_incentive', 'restricted_stock', 'total_payments', 'shared_receipt_with_poi', 'loan_advances', 'expenses', 'from_poi_to_this_person', 'other', 'fraction_from_poi', 'from_this_person_to_poi', 'director_fees', 'to_messages', 'deferral_payments', 'from_messages', 'restricted_stock_deferred']

There are many features so I use PCA to reduce the dimensions. The n-component for each algorithm is determined by experiments of different values.

The new features I created are ‘fraction_from_poi’, and ‘fraction_to_poi’ from original data. A person has high volume of from_poi_to_this_person doesn’t necessarily mean he/she emails poi person more frequently without considering the total received emails amount. So the fraction between emailed from or to poi and total received emails or sent emails is more accurate to track email relationship. I also tested old features using the same clf. The precision and recall score with this new feature are 0.39 and 0.38, and with the old features the scores drop to 0.24 and 0.19.

I scaled the data using min-max algorithm. It’s because I selected email and finance features which have huge difference in scale range. And when using PCA later to reduce the dimension, scale data using min-max is more reasonable. Also for some of the algorithms such SVC and K-means which are scale sensitive, recalling is necessary.

3. I tried three algorithms, and being end up with decision tree. SVC gives a good accuracy but can’t find any true positives. K-means clustering can give both good accuracy and f-1 score on the testing dataset, but when running the tester.py, the recall, precision and f-1 score are lower than .3, which I think is because it’s too dependent on the initial condition so makes it difficult to train.

4. Tuning parameters is essential to get a good model, and it can result in poor accuracy and predictions even using the correct algorithm. For example, I use decision tree, and GridSearchCV to tune min_samples_split with a set of values, and found the best values for this parameter. Another parameter I tune using GridSearchCV is n_components for PCA.
5. Validation is to evaluate the algorithm, and it could cause overfit if validation went wrong. I validate the result using a test data set that randomly picked from the original dataset, by stratified shuffle split cross validation.
6. I use accuracy, precision, recall, and f-1 score as my evaluation metrics. My accuracy by average is above 0.8 for my testing dataset, and precision, recall and f-1 score are around 0.3. Sometimes even I got very low precision and recall scores, the accuracy is always good. This is because most of the suspects are non-poi, and it’s easier to label non-poi. So the high accuracy alone doesn’t mean a good model, and it needs to check precision, recall and f-1 score to tell if the model is good to predict POI.

   Accuracy tells if the label is predicted correctly from the model. The high precision score doesn’t necessarily mean the model can discern between a poi and non-poi. Since most suspects are non-poi, it could end up with high precision score but no or less poi been identified. Precision tells about the percentage of true poi among all predicted poi from the model. The higher the precision, the more accurate for the model to predict a true poi. Recall tells about how much degree the model can identify and recover pois from all true pois. The higher the recall, the more poi have been correctly identified.