Answers are corresponding to the questions in this link

1. The goal of this project is to identify who are the persons of interest of Enron scandal. The dataset contains 146 data points which represents staff of Enron. There are 19 POIs and 127 non-POIs. Each data point has some financial features and email features, such as salary, bonus, stock options and the number of mails with person of interest. But, there are some missing values which are represented by 'NaN'. From the viewpoint of the objective of this project, I thought the number of mails involved with POI is the key point of my analysis. Before I tune up my classifier, I made some scatter plots for each feature and found an outlier which represents the total amount of other data points. So I removed this data point from training dataset.

2. For my final classifier, I used three features and I add one feature by my own. At first, I checked feature importances of all parameters and get rid of some weak parameters. When I use 'bonus', 'restricted_stock', 'salary', 'exercised_stock_options', 'deferral_payments' for the parameter, accuracy was 0.815. Based on feature importances of parameters, I removed 'bonus' and 'restricted_stock', and finally, I used 'salary', 'exercised stock options', 'deferral payments' and 'fraction of messages sent to persons of interest'. Their feature importances are 0.13, 0.38, 0.08 and 0.41. The feature I made was the most important feature for my classifier.

3. The algorithm I used was Decision Tree. I also tested gaussian naive bayes, AdaBoost and Random Forests. The results between algorithms with the features I used was below. Decision Tree algorithm has the highest accuracy of them, and in this big fraud case, I thought that lessen the fraction of false negatives (which means the people who is actually a person of interest but was not considered to) is the most important. So I chose Decision Tree algorithm which has the highest Recall value.

	Accuracy	Precision	Recall	F1
Decision Tree	**0.85722**	0.54009	**0.47690**	**0.50653**
GaussianNB	0.84843	0.51478	0.25780	0.34355
AdaBoost	0.84312	0.48541	0.32770	0.39126
Random Forests	0.85658	**0.58591**	0.23120	0.33156

4. With tuning parameters of algorithms, we can optimize classifier. If we don't tune parameters, we might be influenced by outliers too much and our decision boundary doesn't work well. For Decision Tree Classifier I used for final project, I tried GridSearchCV to tune parameters.

Tested parameters are below.

```
param_grid = {"criterion": ["gini", "entropy"],
       "min_samples_split": [2, 5, 10, 20],
       "max_depth": [None, 2, 5, 10],
       "min_samples_leaf": [1, 5, 10],
       "max_leaf_nodes": [None, 5, 10, 20],
       }
```

I got best performance with criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_samples_leaf=1, min_samples_split=5.

5. Validation is the way how I assess the performance of classifier. The common mistake in validation process is unconsidering overfitting. If I use same dataset between constructing classifier and assessing accuracy, the classifier might be optimized too much for that dataset. This would result poor performance in other dataset. To avoid this problem, we should split the dataset into training and testing sets, and try to optimize indicators with testing dataset. I used train_test_split() function to split original dataset into training and testing by 70% and 30%.

6. I used three evaluation metrics, precision, recall and F1 score. The table above shows evaluation metrics for each algorithm. Precision describes the accuracy of the number of algorithms predicted to POI. Recall describes the rate of the number of predicted to POI over the number of actual POI. So it could be say that if we have high precision, people who were predicted to POI is highly suspicious, and if we have high recall, it means we could predict most of POIs in the dataset. In addition, F1 score means a weighted average of the precision and recall, so it can be overall rating for algorithms.