# Analysis of ENRON Dataset

**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?**

Answer: The goal of this project is to find employees in ENRON who might be involved in corporate fraud based on the publicly available datasets containing financial and email data. This dataset will be useful as it contains the most crucial information about the ENRON's employees which can be used as features in machine learning to predict whether a person might be involved in fraudulent activities or in other words 'is he/she a 'person of interest' (POI)?'.

The dataset contains samples of around 145 employees from ENRON. 18 people have been declared as POI and 127 as non-POI in the dataset. There are 21 features available for each sample. Some of the features are salary, bonus, stock value, total payments and email related information like number of messages received and sent and number of interactions with POIs. Some of the features with large number of missing values are deferral_payments(107 values missing), loan_advances(142 values missing), restricted_stock_deferred(128 values missing), deferred_income(97 values missing) and director_fees(129 values missing). I found an outlier in the dataset with key 'TOTAL' which looks to be summing the values of various features of each employe. I removed it by dropping the corresponding key and item from data_dict.

**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.**

Answer: The features I used are 'rescaled_salary', 'rescaled_to_poi_fraction', 'rescaled_bonus', 'rescaled_total_stock_value', 'rescaled_total_payments'.
To select the features I iterated through various number of features in loop and applying SelectKBest method to pick up the features that contributed to maximum amount of variance in the labels. I ended up choosing 5 features for my algorithm. I did feature scaling on all the features that I used because I wanted to use them in SVC. The new features that I created are 'to_poi_fraction' = 'from_this_person_to_poi'/'from_messages' and 'from_poi_fraction' = 'from_poi_to_this_person'/'to_messages'. I created because I felt that it's the fraction of the total messages sent or received which involved a POI that would help us in deciding whether a person is POI, rather that just an absolute number of messages to and from POI. I used

SelectKBest technique and the scores for the 5 features that I chose are [ 18.86213243 16.87387044  3.29382856  21.3259171  24.75220603
  8.96762538]. For choosing parameter values of decision tree and svc, I used GridSearchCV to iterate through various options for the parameter and to choose the best performing ones returned by the GridSearchCV.

**3. What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms?**
Answer: I chose Decision Tree Classifier. I tried SVM as well. To compare the performance of the model, I relied upon the f1 score of each model in the local data.

**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?**
Answer: The performance of each Machine Learning algorithm varies greatly based on the selection of the parameters and underlying dataset. Therefore, in addition to choosing the right algorithm for the use case, we also need to tune the parameters based on our training data by trying out variety of values for parameters and choosing the ones for which the classifier performs the best on our training and cross validation data. I tuned the parameters using GridSearchCV and in my final Decision Tree Classifier, I used min_samples_split=3, max_depth=5

**5. What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis?**
Answer: Validation is check the performance of our algorithm on sample dataset. One way to validate our model is to hold out certain amount of data from the available training data to serve as evaluation purpose, using which we can select the algorithm, tune the parameters of our algorithm and measure performance. A classic mistake is to use the same set of training data to train and evaluate the model, which may lead to over-optimistic score to our model. Such model may underperform in the new test data, as we might have overlooked the issue of overfitting while using wrong validation technique. I used KFold technique with 10 folds to separate given data in training and validation sets.

**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.**
I used precision, recall and f1 score to evaluate my model during validation phase in each KFold iteration. Precision in our case means the fraction of people who are actually POI out of all the people predicted as POI by our algorithm. Recall means the fraction of people identified as POI out of all the people who are actually POI.
In poi_id.py, The average precision: 0.133, average recall: 0.3, average f1 score: 0.1833

In tester.py the scores are Accuracy: 0.82907  Precision: 0.34707  Recall: 0.32000
F1: 0.33299  F2: 0.32507

Total predictions: 15000 True positives: 640 False positives: 120 False negatives: 1360
True negatives: 11796