**This document gives a brief idea about the steps I took while performing Data exploration and building a predictive model to identify the person of interest in Enron scandal.**

In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, there was a significant amount of typically confidential information entered into public record, including tens of thousands of emails and detailed financial data for top executives. In this project, I played a role of detective, and used my Machine Learning skills to use by building a person of interest identifier based on financial and email data made public as a result of the Enron scandal.

There was a single outlier in the dataset. After visualizing the features I noticed outlier. This was identified during the phase of data exploration. This outlier was removed as part of data cleaning process. This dataset contains 146 data points.

The features I used in my final algorithm is listed as below.

features\_list=['poi','salary','bonus','total\_payments','long\_term\_incentive','expenses']

After trying a lot of features and measuring the performance of various features and different algorithm I choose the above mentioned features. They gave highest number of recall and precision score when compared with other set of features. I also used PCA as part of feature selection process but the score achieved was slightly less than the above used features. I used Min Max Scaler to perform scaling. I also created two new features

['fraction\_from\_poi','fraction\_to\_poi']. I used Decision Tree algorithm in my final analysis.

Also found the feature importance of the features.

I tried with various algorithms like Naïve Bayes, Decision Trees, and K means clustering.

After trying with various email and Financial features, I ended up using Decision Trees as it gave best score of recall and precision scores. The performance of the model created by using Decision Tree was better than other models.

Tuning the parameter of an algorithm helps in increasing the performance of the model. If the parameters of the algorithm are not tuned properly they might decrease the score of your model. So tuning the parameters is like selecting the right things in right proportion to get maximum score/best results and performance. I tuned the following parameters from the decision tree algorithm: criterion, splitter, max\_depth, min\_samples\_split. Observed the performance by trying different values and then set it to the values which gave me best scores.