### **Understanding the Dataset and Question**

**Q.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?  [relevant rubric items: “data exploration”, “outlier investigation”]

**Ans:** The aim of this project is to identify Enron employees who may have committed fraud based on the public Enron financial and email dataset. The provided (financial) dataset contains 21 features for each person. The email dataset contains the email text and author of the emails.There are total 121 data points in dataset amongst which 16 are POI's ( 13.2 %). By using machine learning on this dataset, we can find particular patterns in the data to detect the “person of Interest “, and hence we can detect the frauds involved. There are few (3) outliers in the dataset. I detected them with the help of visualization using matplotlib scatter plot. As all of them were non POI's, I got confused whether to keep them or remove them so I checked the classifier performance with and without removing them, but on the either cases, the classifier performance did not changed much.

**Optimize Feature Selection/Engineering**

**Q.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”, “properly scale features”, “intelligently select feature”]

**Ans:** I have picked 3 features by hand, on trial and error basis. I tried using PCA for feature selection, and it was giving satisfied results in my code, but was giving bad results while running tester.py The features are 'exercised\_stock\_options' , 'to\_messages' , 'conversation\_with\_poi'. The feature 'conversation\_with\_poi' is created by me by adding features 'from\_this\_person\_to\_poi' and 'from\_poi\_to\_this\_person'. Here I thought the person who gets and sends more emails to / from POI's can be himself a POI. I treid scaling features and it worked when I used featues 'to\_messages', 'exercised\_stock\_options' , 'deferral\_payments'. But the current set of features gave me the best results compared to these scaled ones.

**Pick and Tune an Algorithm**

**Q.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”]

**Ans:** In total I tried 6 algorithms. These are: 1. Naive bayes, 2. DecisionTree, 3. Support vector Machine, 4. K Nearest Nieghbors, 5. Random Forest, 6. Adaboost. SVM and Random forest didn't worked well and they took more time for training compared to other algorithms. Amongst all of these, KNN worked well compared to the others. KNN resulted in the precision of 0.57, recall of 0.39 and f1-score equal to 0.46. The classifier algorithm also gave the good accuracy of 0.86.

**Q.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? (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 item: “tune the algorithm”]

**Ans:** Tuning the parametrs ofan algorithm means adjusting the algorithm's decision boundary as per the data and finding the best decision boundary by using the trade-off between the bias and variance. For tuning parameters, I have used GridSearchCV from sklearn. The 'n\_neighbors' and 'p' are the parameters I have used for tuning the KNN algorithm. Here 'n\_neighbors are the number of closest data points (labels in the training data ) used by the classifier for deciding, to which class the current observation belongs to, based on the distance metric. 'p' is used to determine which distance metric the algorithm should use while training. P=1 means manhattan distance and p=2 means euclidean distance. I have also used the weights = 'distance' as a default parameter in case of tie. When tie happens between the number of closest points to the current obsservation, the classifier will choose consider the most closest point's labels if I mention weights = 'distance'.

**Validate and Evaluate**

**Q.5** What is validation, and what’s a classic mistake you can make if you do it wrong? How did you validate your analysis?  [relevant rubric item: “validation strategy”]

**Ans:**  Validation means validating our algorithm's performance with the new data, which is not used for the training of an algorithm. Therefore we split our dataset into 2 sets, that are training set and testing set. We use training set along with the training labels to train out classifier algorithm and we validate it's performance by doing prediction on testing set. If we do not do validation, that is we do not split our data into training and testing set then the mistake we commit is we test our algorithm using the same data which is used for the training of an algorithm, hence we are not validating our algorithm's performance. Due to this our algorithm may give a very good training and testing accuracy but may fail for the new observations. Therefore it is veryimportant that we do validation of our classifier by splitting our dataset into training and testing sets.

I have used sklearn's train\_test\_split() validation strategy for splitting the dataset into training and testing sets with the test\_size = 0.3, whic means my testing dataset is 30% of the total dataset.

This method has worked very well and gave good results.

**Q.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”]

**Ans:** To evaluate the results of the classifier algorithm, I haveusedtheevaluationmetrics precision, recall and F1 score. My final algorithm, that is KNN, resulted in the precision of 0.57, recall of 0.39 and f1-score equal to 0.46. It also gave the good accuracy of 0.86.

Here precision of 0.57 means amongst the all predicted labels as POI's by the algorithm, 57% of them are successfully identified by the algorithm as true POI's (true positives). The recall of 0.39 means amongst the all POI's present in the dataset, 39% of them are predicted correctly by an algorithm.