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”]

   The goal of this project is to identify a given person is a person of interest or not based information on each person. The data set contains the financial and email information of each person. Given data is used to train multiple classifiers and test the performance of each of those on test datasets.

   After examining the dataset, by plotting, there was one outlier found which is a record for total of all data points. The data point was removed from the dataset before further processing.

   Dataset structure:

   No. of training examples (with outlier) – 146

   No. of training examples (outlier removed) – 145

   No. of POI data points – 18

   No. of Non-POI data points – **127**

   No. of features used for training – **9**

   Dataset structure of Email training:

   No. of POI emails – **18**

   No of non POI emails – **18** (random sample on 127emails)

   Total # of training examples - **7634**

   Total no. of features pre-PCA – **32930**

   Total no. of features post-PCA – **1000**

   No. of training examples – **6107**

   No. of test examples - **1527**

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”]

Features Used:

Featured used in the training

['salary','total_payments','from_this_person_to_poi','from_poi_to_this_person','shared_receipt_with_poi','bonus','total_stock_value','from_poi_frac','to_poi_frac']

Features used are have mix of email and finance data of each employee. Most important features are salary, from_poi_frac, to_poi_frac.

New Features:

New features created – from_poi_frac, to_poi_frac

from_poi_frac -> fraction of emails received from poi to total emails received.

to_poi_frac -> fraction of emails sent to poi to total emails sent.

Feature Scaling:

There was a minmax scaler applied to scale all features to be in the range of 0-1, since the range of feature values had very high difference between them, like Salary and from_poi_frac for instance.

Feature Importance:

Below are some of the most important features when fit by a decision tree classifier.

to_poi_frac, shared_receipt_with_poi, total_payments

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”]

Algorithms used – Linear SVM, Gaussian SVM, Decision Tree, Gaussian Naïve Bayes.

Based on performance of different algorithms , Gaussian SVM performed best on the shuffled data.

Although when the Gaussian SVM was applied on the dataset with out the shuffle split it did not provide good results.

Decision tree worked best when the classifier was applied on the split of dataset with out the shuffle split.

Email processing and training:

Additionally, emails of all employees has been processed using text learning, vectorization and a Gaussian SVM has been used to train and evaluate metrics.

PCA has been applied and the dimension of features were reduced to 1000, since it had lot of features to process.

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"]

Parameter tuning means modifying the way the data is trained by an algorithm. By modifying the parameters, the fit can be made high biased or high variance. A successful parameter tuning aims to find a balance between the bias and variance.

A model with high recall is a high variance – low bias model (Does not generalize well for untrained data)

A model with high precision is a high bias and low variance model. (Generalizes well but miss placing data in the right class more often)

In this experiment I tried a higher min_samples_split(minimum samples needed to further proceed with the split) than the default value of 2, but it did not give best metrics than the default case

In the case of SVM I tried below parameters

Kernel = Linear, Gaussian

C values = [100000, 110000]

Iterations limit = [15000,20000,90000]

Best set of parameters has been chosen based on the test run on above different parameters.

In case of email processing to identify the POIs, PCA was used to tune the heavy number of features to a 1000.

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"]

Validation is the process of checking the performance of a training algorithm on an independent dataset separate from the training dataset.

One big mistake on validation can be evaluating a training algorithm's performance using the same data it was trained on. Doing this will not only give a very optimistic metrics because it was specifically trained for the dataset, it also do not infer if an algorithm is over fitted/Very high variance.

Introducing the classifier to some new untrained data gives the true performance of a classifier and also detects overfitting and make sure the classifier generalizes well for untrained data.

In this experiment the dataset was split in to train and test dataset using a stratified shuffle split (since the training example was very limited)

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"]

Gaussian SVM with C = 100000, iteration = 20000

Precision: 0.38355

Algorithms ability to accurately match a positive example with the truth positives, among all the positive predictions.

Recall: 0.34750

Algorithms ability to predict/recall all the positive examples correctly.

F1: 0.36464

A score calculated using the precision and recall metrics.

Decision Trees with min_samples_split = 2

Precision: 0.37434

Recall: 0.35600

F1: 0.36494

Metrics on Email training algorithm

Precision: 0.98

Recall: 0.97

F-Score 0.97

The algorithm trained on emails of poi and non-poi had excellent metrics, both precision and recall scores were excellent and it worked well on different test samples.