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 the project was to build a classifier to analyse the financial and email information of enron employees and to identify persons of interest. Our dataset consisted of 144 individuals with their person of interest label(poi or not), 14 financial features, and 6 email features including the email address. The dataset is highly unbalanced with only 18 individuals belonging to poi class and 126 belonging to the other class. The project attempts to use these 20 measures and additional derived measures to help build as system which can classify employees based on the measures into poi or not. Whenever data was missing, it was replaced with a zero value. We use machine learning to help identify importance and relevance of the various measures in determining an employees poi status.

Initial analysis of the dataset using scatter plots pointed to an outlier which corresponded to “Total” i.e. total across each financial measures for all employees. This row was removed from the financial dataset when building the classifier.

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”, “intelligently select features”, “properly scale features”]

Based on intuition and some knowledge of financial measures, 7 additional features were created from the initial 19 features (email address was not used as a feature in building the classifier). A MinMaxScaler function was used to limit the values within 0 and 1 across all measures. We used SelectKBest function in sklearn to identify the key features. The MinMaxScaler, SelectKBest was combined with four different algorithms to form four pipelines. Grid Search CV was used to choose the best combination of hyperparameters. The best value for K turned out to be 12 and the algorithm with best F1 value was Gaussian Naive Bayes. The top 12 features from the GridSearch is given below.

- salary
- total_payments
- loan_advances
- bonus

- deferred_income
- total_stock_value
- exercised_stock_options
- long_term_incentive
- restricted_stock
- shared_receipt_with_poi
- shared_receipt_with_poi_to_messages
- to_poi_to_messages

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”]

Four algorithms were tested to compare performance of the classifier - GaussianNB, DecisionTreeClassifier, LogisticRegression and KNeighbors Classifier. KNeighbors classifiers was unable to identify the positive cases and were therefore rejected. Between the remaining three, GaussianNB had better Precision , Recall and F1 score and therefore was chosen as the algorithm to build the classifier.

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? What parameters did you tune? (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 items: “discuss parameter tuning”, “tune the algorithm”]

Every algorithm has a few parameters called hyperparameters, than can be used to modify the performance of the algorithm. By changing the parameters, the algorithm can be tuned to provide best performance possible for a given dataset. In case of tuning the algorithms, GridSearchCV was used to test across a range of values for the hyperparameters (optimal k for SelectKBest etc) and the best performing parameters are chosen for the final model. Our final model has a GaussianNB algorithm that does not lend itself to hyperparameter tuning. In case of DecisionTreeClassifier we tuned min_samples_split parameter using GridSearchCV. We tuned the C value for LogisticRegression Classifier and the number of neighbors parameter for KNeighbors Classifier.

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 items: “discuss validation”, “validation strategy”]

Validation is the process of splitting the entire training data into a train set and a test set, training the algorithm on the train set and testing on the test set to check for performance. If we do not make this separation and instead train and test on the same dataset, overfitting will take place.

The resulting algorithm may work well on the training dataset but not work well with new datasets.

The results also change depending on how the data is split. So to get an indication of average performance, we can conduct cross validation. By using GridSearchCV we have employed 3-fold cross validation on our dataset. The train data is further split into two segments - train and test and GridSearch is done by training on this new train data and testing on the new test data. Then the process is repeated using a new way to split the training data into train and test. In case of 3-fold cross validation the process is repeated a total of 3 times. The displayed results are the average across the three attempts. Thus this gives a better indication of the average performance of this algorithm and is not unusually influenced by the way the data is split.

In the final step, to evaluate the performance of our chosen algorithm we train using the entire training data and test on our testing dataset.

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"]

The two evaluation metrics used are precision and recall. Precision measures what percentage of the Persons of Interest have indeed been identified as a 'poi'. Recall measures what percentage of individuals identified as 'poi' are indeed persons of interest.

In case of our optimal classifier - Gaussian Naive Bayes, Select K Best with K = 12 and MinMaxScaler, the average precision is 0.86 and average recall is 0.86.