POI Identifier

- 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 a system that can classify employees based on the features 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 features 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. A scan through the employee names showed “The Travel Agency in the Park” listed as an employee. Since it was clearly not the name of an employee, the entry was removed. removed. The employee ‘Lockhart Eugene E” had “Nan” value for all given features. Therefore this entry was also removed. The features Loan Advances and Director Fees have very few positive values. But Loan Advances appear non zero for a few POI’s and so is retained. Director Fees are much smaller in magnitude compared to payments and does not display any major relationship to POI status. This feature was excluded from further analysis.

- 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, 5 additional features were created from the initial 18 features (email address and Director fees were not used as a feature in building the classifier).

They are:

1. Deferred_income_to_total_payments - Measures the proportion of the payments received by an individual that is deferred. If the individual has more deferred income, then he has an incentive to ensure the company survive. So a higher value is less indicative of POI association.
2. Exercised_stock_option_to_stock_value - The proportion of stock value through exercised stock option. If an individual believes the company may not perform well in future, then he has the incentive to exercise stock options,making it a possible indicator of POI.
3. Shared_receipt_with_poi_to_messages - This scales the emails shared with POI based on the total emails received by an individual. If a person is in position like HR and receives all emails in the company, he will have the highest value for shared receipt with POI as well as the highest number of 'to' emails. This feature helps eliminate such individuals
4. To_poi_to_messages - The proportion of emails sent to POI by an individual to all emails sent by him .If an individual is a POI, this ratio will be higher for him even if the emails sent to POI is not as high as someone whose position requires him to send a lot of emails.
5. From_poi_to_messages - The proportion of emails received from POI by an individual to all emails received by him. If an individual is a POI, this ratio will be higher for him even if the emails received from POIs are actually not high.

The last three measures help refine the relationship an individual has with POIs by relating the communication an individual has with POIs' to the total communication the individual has with all employees.

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 5 and the algorithm with best F1 value was Gaussian Naive Bayes. The scores for all features from the Select K Best is given below. The chosen 5 features are in bold letters. Two of the derived features are in the top 6 based on the feature scores.

Original Features

- **salary ---15.859***
- deferral_payments --- 0.01
- total_payments --- 8.959
- loan_advances --- 7.038

- **bonus --30.729***
- restricted_stock_deferred -------------------------------- 0.727
- deferred_income --- 8.792
- **total_stock_value --10.634***
- expenses --- 4.181
- exercised_stock_options ---------------------------------- 9.68
- other -- 3.204
- long_term_incentive -- 7.555
- restricted_stock -- 8.058
- to_messages -- 2.616
- from_poi_to_this_person --------------------------------- 4.959
- from_messages -- 0.435
- from_this_person_to_poi --------------------------------- 0.111
- **shared_receipt_with_poi ----------------------------------10.723***

Derived Features

- deferred_income_to_total_payments ------------------ 0.155
- exercised_stock_option_to_stock_value -------------- 0.639
- shared_receipt_with_poi_to_messages ---------------10.618
- **to_poi_to_messages -------------------------------------15.838***
- from_poi_to_messages ------------------------------------ 0.519

*Selected Features

- 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 on the test data and therefore was chosen as the algorithm to build the classifier.

Comparison of Classifier Performance

Classifiers	**Precision**	**Recall**	**F1_score**
GaussianNB	0.89	0.88	0.89
DecisionTreeClassifer	0.80	0.74	0.77
LogisticRegression	0.83	0.86	0.84
KNeighborsClassifier	0.78	0.86	0.82

- 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.
The following values of k - 5,8,10,12,15,20,25 were tested for tuning SelectKBest feature selection.

We ran four different algorithms to select the optimal 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.

- DecisionTreeClassifier: min_samples_leaf' - 1,2,3; min_samples_split - 2,3,4
- LogisticRegression: C - 0.01,0.1,1,5,10,5
- KNeighborsClassifier; n_neighbors - 1,3,5,7,9

- 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 data was split into training and test datasets. GridsearchCV was used to determine the optimal hyperparameters for each algorithm based on the training data. The best estimator for each algorithm was applied to the test data to compare the precision and recall as well as F1 rates on the test data. The algorithm giving best results on the test dataset was chosen as the final model. Thus by training on one dataset and testing on the held out dataset, we were able to overcome overfitting.

- 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 the percentage of individuals identified as 'POI' who are indeed persons of interest. Recall measures the percentage of the Persons of Interest who have indeed been identified as a 'POI'.

In case of our optimal classifier - Gaussian Naive Bayes, Select K Best with K = 5 and MinMaxScaler, the average precision is 0.89 and average recall is 0.88.