UD120 Intro to Machine Learning - Final Project

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 build a model which can be used to identify persons of interest at Enron from available data. Persons of interest describes employee's at Enron who may have been involved in corporate fraud activities. The dataset encompasses financial and email data plus a POI flag which can be used as a target variable for all directed classification algorithms. Machine learning is a powerful tool to uncover patterns in the data which may not be immediately obvious to a human due to the scale and/or complexity of the data.

There are 146 records in the dataset, but of these "TOTAL" and "THE TRAVEL AGENCY IN THE PARK" are not actual employees of Enron and so have been removed from the analysis. Other outliers appear to be indicative of POIs rather than errors which need to be removed. Of the 144 people, 18 have been flagged as POIs. There are 20 features. The financial features are in US dollars, ***email_address*** is a text string and the other email features are counts. No one feature has a value for everyone. ***loan_advances*** has the fewest populated values (3), second least populated is ***director_fees*** (16), third least is ***deferral_payments*** (38), and the most populated is ***total_stock_value*** (125).

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”]

New features were created to track what percentage of a person’s email were sent to a person of interest and what percentage were from a POI (***fraction_from_poi*** and ***fraction_to_poi***). The intuition being that a greater correspondence with POIs is indicative of being a POI themselves. I have also created a new feature ***has_email_address*** which is set to 1 when a person does have an email address and 0 otherwise. All POIs in dataset have an email address, so the absence of one may indicate a person who can be safely marked as non-POI. The ***email_address*** feature was then removed as it does not provide any further useful information.

I selected features using a decision tree classifier which identified 9 features which had a non-zero importance in identifying POIs. Of these, the 4 lowest importance features were found to have a detrimental impact on the classification algorithms (both performance and results) so the threshold was raised from 0 to 0.1. Because not all features are of the same scale and some of the classification algorithms I am evaluating calculate Euclidean distances, the features were also scaled using a ***MinMaxScaler***. The selected features are:

Feature	**Importance**
exercised_stock_options	0.216
other	0.190
expenses	0.170
fraction_to_poi	0.136
shared_receipt_with_poi	0.119
~~total_stock_value~~	~~0.056~~
~~total_payments~~	~~0.042~~
~~bonus~~	~~0.042~~
~~restricted_stock~~	~~0.028~~

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”]**

5 algorithms were evaluated. The selected algorithm is highlighted in red below:

Algorithms	**Accuracy**	**Precision**	**Recall**	**F1**	**F2**
Naive Bayes (GaussianNB)	0.838	0.361	0.176	0.237	0.196
Support Vector Machines (SVC)	n/a	n/a	n/a	n/a	n/a
K-Nearest Neighbours (KNeighborsClassifier)	0.872	0.904	0.114	0.202	0.138
AdaBoost ensemble (AdaBoostClassifier)	0.881	0.622	0.424	0.504	0.453
Decision Tree (DecisionTreeClassifier)	0.868	0.550	0.429	0.482	0.449

The ***AdaBoost ensemble*** algorithm was found to have the best performance. It had the highest Accuracy, F1 and F2 scores. The only other algorithm which passed the 0.3 threshold on Precision and Recall was the ***Decision Tree***, but this had a lower F1 score (a composite of Precision and Recall). Of the other algorithms, ***K-Nearest Neighbours*** scored highest on Precision, but had the lowest Recall, F1 and F2 scores. The ***Support Vector Machines*** algorithm did not return scores due to a lack of true positive predictions. The major downside of AdaBoost ensemble is it is by far the slowest of the algorithms to tune and test.

Algorithms	**Total runtime (seconds)**
Naive Bayes (GaussianNB)	2.890
Support Vector Machines (SVC)	6.030
K-Nearest Neighbours (KNeighborsClassifier)	399.905
AdaBoost ensemble (AdaBoostClassifier)	6874.821
Decision Tree (DecisionTreeClassifier)	113.747

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"]

Most machine learning algorithms accept a number of parameters which control different aspects of their execution, for example which specific method to use, or how "far" to keep going with recursive actions. If the parameters aren't well tuned than this can affect the accuracy of results. For the selected ***AdaBoost ensemble*** method the following parameters were tuned using ***Grid Search Cross Validation***. The selected parameter values are highlighted in red below:

Parameter	**Values**
n_estimators	10, 20, 30, 40, 50, 60, 70, 80
learning_rate	0.5, 1.0, 1.5, 2.0
algorithm	SAMME, SAMME.R

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 an important step in Machine Learning. If we don't validate the results – or only evaluate using the training data - then we can find we have "overfitted" the model to the training data and will not get the same level of accuracy against any new data. Since we have a small dataset to get the most out of the data we do have, this model was evaluated using cross validation. The particular method was a modification of the ***test_classifier*** function in ***tester.py*** which uses a ***Stratified Shuffle Split*** to return 1,000 stratified randomized folds, and then averages results across the folds. This was combined with the aforementioned ***GridSearchCV*** to tune and test each algorithm in turn.

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"]**

When evaluated using ***tester.py*** the selected ***AdaBoost ensemble*** method produced average precision and recall scores of 0.622 and 0.425 respectively. Both are well above the thresholds of 0.3 set for this project. In practical terms this means that about 62% of the time when it identifies a POI it does so correctly (849 true positives vs 517 false positives), and about 42% of the time, any actual POIs will be correctly identified as such (849 true positives vs 1,151 false negatives).