Identity Fraud from Enron Email

1. The main goal of this task was to build a classifier to discover Persons of Interest (POI) using financial and email data from Enron's top executives, made public during the investigation on widespread fraud in the company. Machine Learning is useful not only to handle large amounts of data faster than other algorithms, but also one of its strengths is to find patterns and relations that would otherwise be missed.

   For that, we will use a labeled dataset containing email and financial information from 145 executives of the company, of which 18 are deemed POI's and 127 not. The dataset contains also the sum total of every numerical column. One important feature of this dataset is the large number of missing values. There are 1358 missing values on the dataset, approximately 44% of the total, and 20 features. One clear outlier in the data is the TOTAL row (it's not an outlier per se, but it shouldn't be part of the final set). The outliers were all treated the same way: find out which samples have values above the 95th percentile in at least 8 features (40%).

2. Due to the large number of missing values, one can try to select the relevant features based on the number of missing values for each. The **email_address** was removed from the beginning since it's not a numerical column. A reasonable cutoff is to use the number of actual values for each feature (not missing). Different tests were used with different cutoffs, using a Random Forest Classifier on a test sample to check for the results. Due to the heavily imbalanced dataset, the F1 score was used to gauge the impact of removing features. All of them were scaled to the range [0,1] using a *MinMaxScaler* since the range of the features are very different, reaching sometimes tens of thousands. The table below report the F1 scores for for different cutoffs.

Cutoff	F1 score
50	0.39
60	0.40
70	0.41
80	0.39
90	0.27
100	0.26

Features with at least 70 values were used, resulting in 12 features: **salary, to_messages, total_payments, exercised_stock_options, bonus, restricted_stock, shared_receipt_with_poi, total_stock_value, expenses, from_messages, other and from_poi_to_this_person**.

Missing data can happen for a lot of reasons; however, this being a corporate fraud investigation, one reason for missing data could be due to the numbers not being divulged, or being hidden for some reason. This could, in principle, be a feature of interest for the identifier. Therefore, a new feature called **missing data** was created that counts the number of missing values per person. This ended up being used in the final analysis, together with the ones mentioned above. Below we report the feature importances given from a Random Forest Classifiers.

Salary	to_messages	total_payments	exercised_stock_options	Bonus	restricted_stock
0.098	**0.034**	**0.1**	**0.043**	**0.15**	**0.069**

shared_receipt_with_poi	total_stock_value	Expenses	from_messages	Other	from_poi_to_this_person	Missing data
0.058	0.05	0.14	0.026	0.16	0.058	0.017

3. Three algorithm were tested: Random Forest (RF), Support Vector Machine (SVM) and Gradient Boosting. The latter had the worst performance by far, with very low Recall and F1 score. While the Recall score for the SVM was good, it had the worst precision of the three of them. In contrast, the RF had the best precision score and a middling recall score, which, when combined, gave the best F1 score. It is important to notice that RF was also the slowest method to train and optimize by a good margin.

4. Tuning an algorithm means finding the hyperparameters that optimize a given metric. Hyperparameters cannot be learned from the data, and are set before the training begins: it is important therefore to utilize some means of optimization to make sure the best model for that specific task is being used. Not tuning the parameters of an algorithm means that your model will not be optimal for the dataset you are using, and the performance will suffer because of it.

   For the Random Forest, a Randomized search was used to tune both the number of estimators (random integer between 400 and 200), the max depth of the forest, (random integer between 2 and 10), and the minimum number of samples for a node to be split (random integer between 2 and 6). Since the main aim of this task is to obtain a good precision and recall, the tuning was done to maximize the F1 score. A Grid search was also used for testing, but the best results were found when the parameters were randomly drawn from given distributions.

5. A validation set is used to provided an unbiased estimate of the model fit on the training set, and it’s also used to tune the hyperparameters. If validation is not done, the model might not perform well on unseen data, despite having very good performance during training (overfitting). The training and validation sets have to be chosen carefully, so that there’s a representation of both classes in both sets.

   A cross validation was performed with a repeated Stratified Kfold, with 10 repeats and 5 folds each, which amounts to testing the model in 20% of the data for each repetition.

6. Given the highly imbalanced dataset, where only 18% belong to the positive class, accuracy is not a very good metric to look for: a classifier that predicts only the majority class would achieve an accuracy of approximately 88%. In these cases, using a different metric is needed to evaluate the performance of the model.

   Here, we will use **Precision and Recall** as the evaluation metrics for this problem.

   Recall measures the completeness of the classifier. It attempts to answer *What proportion of the given class was correctly identified?*. Its value is given by the ratio of correctly predicted classes and the total number of actual members of the class. A low recall value in this case means that the classifier is having problems identifying POI’s.

Precision measures how efficient the classifier is. It attempts to answer the question *What proportion of the predicted labels are correct?*. Its value is given by the ratio of correctly predicted classes and the total predictions for that class. In this case, a low precision value means that the classifier is mislabelling a lot of the POI's as regular people.

The values for these metrics, plus the F1 score (the harmonic mean between precision and recall) are reported below for the optimised Random Forest model.

Precision	**Recall**	**F1**
0.34	0.49	0.40