Bryony Miles - Enron Project Questions Answered (more details in the report)

Question: Which Enron Employees may have committed fraud based on the public Enron financial and email dataset.

Data: 146 entries, 21 features (financial, email related, **poi**)
poi = individuals who were indicted, reached a settlement or plea deal with the government, or testified in exchange for prosecution immunity

Machine learning is particularly useful in this case as the entries fall into two categories: guilty (poi) or not guilty (non-poi).

During the data exploration I found:

- various NaN values which I analysed and converted to 0
- two invalid entries - TOTAL and THE TRAVEL AGENCY IN THE PARK which I deleted
- two data entry issues - BHANTNAGER SANJAY and BELFER ROBERT which I updated using the original data
- I also looked at two POIs with a lot of nulls and 20 POIs with very high entries but decided that the data was relevant and kept them in.

The features I ended up using mostly composites:

- *key_payments:* salary + bonus + other
- *deferral_balance:* deferral_payments + deferred_ income
- *retention incentives:* long_term_incentive + total_stock_value
- *total_of_totals:* total_payments + total_stock_value
- loan_advances

I ran SelectKBest and SelectPercentile on the original features but the results were very inconclusive and I decided to hand pick the features instead.

I scaled the data with a MinMaxScaler as the amounts varied considerable between features. I experimented with scaling before and after the train/test split. With my final algorithm the scaling after the split worked best.

After testing five algorithms (GaussianNB, LinearSVC, Decision Tree, Logistic Regression and Random Forest) I ended up with Random Forest Classification. You can see the results for the other algorithms in the report.

It took a bit of parameter tuning and playing with the features and train/test split to get it right but I got there in the end. The evaluation metrics I was looking at were precision and recall. Initially the Random Forest Classifier had fantastic precision (0.6) and ok recall (0.23). Here is the final classifier and results.

```
clf = RandomForestClassifier(n_estimators=100, min_samples_split=4,
max_features=None,oob_score=True)

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features=None, max_leaf_nodes=None,
            min_samples_leaf=1, min_samples_split=6,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
            oob_score=True, random_state=None, verbose=0, warm_start=False)
    Accuracy: 0.86947  Precision: 0.51789 Recall: 0.30400   F1: 0.38311 F2:
0.33137
    Total predictions: 15000    True positives:  608   False positives:  566
False negatives: 1392  True negatives: 12434
```