Project 5 Identify Fraud from Enron Email

Reference

https://wiki.python.org/moin/UsingPickle

https://docs.python.org/2/library/time.html

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html

Answers to the questions

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 identify fraud from Enron data.
    - This data set contains various payments, people of interest, emails information. There is an outlier called ‘Total’ when I was exploration the data. I removed this data. Training the classifier used for this dataset, can help us to obtain the best accuracy, precision and recall values, based on the Enron data set, to predict the fraud.
    - Yes, there is one outlier in the data set, which is ‘Total’. I found that this is an outlier as the relevant values stored in this key is too high, and illogically, by looking to the values of each key’s values.
    - I removed this outlier.

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

- The features that are selected are: salary, total_payments, loan_advances, bonus, deferred_income, total_stock_value, expenses, exercised_stock_options, other, long_term_incentive, restricted_stock, from_poi_to_this_person, shared_receipt_with_poi, and to_poi_message_ratio.
- I used the SelectKBest method of scikit-learn.
- I didn’t do any scaling.
- Because there was no need to.
- I make two new features, which is to_poi_ratio and from_poi_ratio, which measures frequently a person send and receive message from POI out of all the messages of his own.
- They were not selected.

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”]

    - I used two algorithms, which are Random and Adaboost, respectively.
    - I also tried GaussianNB algorithm.
    - The accuracies of both of the algorithms are similar, whilst the Random algorithm gives a higher precision when the number of features are less than 10. They show close recall ability when the number of features are less than 10, with the AdaBoost algorithm winning out when the number of features go beyond 10.

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? (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 item: “tune the algorithm”]

- Basically, to tune the parameters of an algorithm is to optimize those parameter, so as to help the algorithm achieve the best result. If we don't do this well, the calculated values via this algorithm will not be good.
- In my case, I used GridSearchCV to find the optimized parameters of the classifiers that I took, for the highest recall score they can achieve.

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 item: "validation strategy"]

   - Validation is what we use to verify the performance of our tuned algorithms on the specific data set. If we don't perform the validation correctly, the classic mistake we would make is to use algorithm/model which is not suitable or good enough for our data set, and consequently lead us to the wrong judgement of the data.
   - In my case, I used 3 metrics, accuracy, precision and recall, on both of the algorithm, Random, and the algorithm, AdaBoost, for comparison.

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"]

   Given in the following table, I listed the metrics comparison between the algorithm Random and algorithm AbaBoost, in accuracy, prevision, and recall, as I mentioned in the last question.

Classifier	Accuracy	Precision	Recall
Random	0.859	0.562	0.392
AdaBoost	0.800	0.332	0.300

There is no doubt that, after carefully tuned, the algorithm Random, shows higher prediction ability in each of the 3 categories, and thus should be selected as the best algorithm.