Project Introduction

In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, there was a significant amount of typically confidential information entered into public record, including tens of thousands of emails and detailed financial data for top executives. In this project, you will play detective, and put your new skills to use by building a person of interest identifier based on financial and email data made public as a result of the Enron scandal. To assist you in your detective work, we've combined this data with a hand-generated list of persons of interest in the fraud case, which means individuals who were indicted, reached a settlement, or plea deal with the government, or testified in exchange for prosecution immunity.

Pep Talk

A note before you begin: the projects in the Intro to Machine Learning class were mostly designed to have lots of data points, give intuitive results, and otherwise behave nicely. This project is significantly tougher in that we're now using the real data, which can be messy and doesn't have as many data points as we usually hope for when doing machine learning. Don't get discouraged--imperfect data is something you need to be used to as a data analyst! If you encounter something you haven't seen before, take a step back and think about a smart way around. You can do it!

Resources Needed

You should have python and sklearn running on your computer, as well as the starter code (both python scripts and the Enron dataset) that you downloaded as part of the first mini-project in the Intro to Machine Learning course. The starter code can be found in the final_project directory of the codebase that you downloaded for use with the mini-projects. Some relevant files:

poi_id.py : starter code for the POI identifier, you will write your analysis here

final_project_dataset.pkl : the dataset for the project, more details below

tester.py : when you turn in your analysis for evaluation by a Udacity evaluator, you will submit the algorithm, dataset and list of features that you use (these are created automatically in poi_id.py). The evaluator will then use this code to test your result, to make sure we see performance that's similar to what you report. You don't need to do anything with this code, but we provide it for transparency and for your reference.

emails_by_address : this directory contains many text files, each of which contains all the messages to or from a particular email address. It is for your reference, if you want to create more advanced features based on the details of the emails dataset.

Steps to Success

We will provide you with starter code, that reads in the data, takes your features of choice, then puts them into a numpy array, which is the input form that most sklearn functions assume. Your job is to engineer the features, pick and tune an algorithm, test, and evaluate your identifier. Several of the mini-projects were designed with this final project in mind, so be on the lookout for ways to use the work you've already done.

The features in the data fall into three major types, namely financial features, email features and POI labels.

- financial features: ['salary', 'deferral_payments', 'total_payments', 'loan_advances', 'bonus', 'restricted_stock_deferred', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees'] (all units are in US dollars)
- email features: ['to_messages', 'email_address', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi'] (units are generally number of emails messages; notable exception is ‘email_address’, which is a text string)
- POI label: [‘poi’] (boolean, represented as integer)

You are encouraged to make, transform or rescale new features from the starter features. If you do this, you should store the new feature to my_dataset, and if you use the new feature in the final algorithm, you should also add the feature name to my_feature_list, so your coach can access it during testing. For a concrete example of a new feature that you could add to the dataset, refer to the lesson on Feature Selection.

Final Project Evaulation Instructions

When you're finished, your project will have 2 parts: the code/classifier you create and some written documentation of your work. Share your project with others and self-evaluate your project according to the rubric here.

Before you start working on the project: Review the final project rubric carefully. Think about the following questions - How will you incorporate each of the rubric criterion into your project? Why are these aspects important? What is your strategy to ensure that your project "meets specifications" in the given criteria? Once you are convinced that you understand each part of the rubric, please start working on your project. Remember to refer to the rubric often to ensure that you are on the right track.

Items to include when sharing your work with others for feedback:

Code/Classifier

When making your classifier, you will create three pickle files (my_dataset.pkl, my_classifier.pkl, my_feature_list.pkl). The project evaluator will test these using the tester.py script. You are encouraged to use this script before checking to gauge if your performance is good enough. You should also include your modified poi_id.py file in case of any issues with running your code or to verify what is reported in your question responses (see next paragraph).

Documentation of Your Work

Document the work you've done by answering (in about a paragraph each) the questions found here. You can write your answers in a PDF, Word document, text file, or similar format.