1. Project Submission

Play detective and put your machine learning skills to use by building an algorithm to identify Enron Employees who may have committed fraud based on the public Enron financial and email dataset.

1. **General Submission and Evaluation Overview**

Your submission will contain several files: the code/classifier you create and some written documentation of your work. We will evaluate your project according to the rubric [**here**](https://review.udacity.com/#!/projects/3174288624/rubric); only projects that satisfy **all** "meets expectations" items will pass. **Please self-evaluate before you submit!** If you don't think your project meets all the criteria, the project evaluator likely won't either.

1. **Submission**

Ready to submit your project? Go back to your Udacity Home, click on the project, and follow the instructions to submit!

* You can either send us a GitHub link of the files or upload a compressed directory (zip file).
* Inside the zip folder include a text file with a list of Web sites, books, forums, blog posts, GitHub repositories etc that you referred to or used in this submission (Add N/A if you did not use such resources).

It can take us up to a week to grade the project, but in most cases it is much faster. You will receive an email when your submission has been reviewed.

If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com.

1. **Items to include in submission:**
2. **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 submitting 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). Notably, we should be able to run poi\_id.py to generate the three pickle files that reflect your final algorithm, without needing to modify the script in any way. If you have intermediate code that you would like to provide as supplemental materials, it is encouraged for you to save them in files separate from poi\_id.py. If you do so, be sure to provide a readme file that explains what each file is for.

1. **Documentation of Your Work**

Document the work you've done by answering (in about a paragraph each) the questions found [**here**](https://docs.google.com/document/d/1NDgi1PrNJP7WTbfSUuRUnz8yzs5nGVTSzpO7oeNTEWA/pub?embedded=true). You can write your answers in a PDF, text/markdown file, HTML, or similar format. The responses in your documentation should allow a reviewer to understand and follow the steps you took in your project and to verify your understanding of the methods you have performed.

1. **Text File Listing Your References**

A list of Web sites, books, forums, blog posts, github repositories etc. that you referred to or used in this submission (add N/A if you did not use such resources). Please carefully read the following statement and include it in your document “I hereby confirm that this submission is my work. I have cited above the origins of any parts of the submission that were taken from Websites, books, forums, blog posts, github repositories, etc.

1. **Good Luck!**

Quality of Code

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Functionality | Code reflects the description in the answers to questions in the writeup. i.e. code performs the functions documented in the writeup and the writeup clearly specifies the final analysis strategy. |
| Usability | poi\_id.py can be run to export the dataset, list of features and algorithm, so that the final algorithm can be checked easily using tester.py. |

Understanding the Dataset and Question

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Data Exploration (related mini-project: Lesson 5) | Student response addresses the most important characteristics of the dataset and uses these characteristics to inform their analysis. Important characteristics include:   * total number of data points * allocation across classes (POI/non-POI) * number of features used * are there features with many missing values? etc. |
| Outlier Investigation (related mini-project: Lesson 7) | Student response identifies outlier(s) in the financial data, and explains how they are removed or otherwise handled. |

Optimize Feature Selection/Engineering

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Create new features (related mini-project: Lesson 11) | At least one new feature is implemented. Justification for that feature is provided in the written response, and the effect of that feature on the final algorithm performance is tested. The student is not required to include their new feature in their final feature set. |
| Intelligently select features (related mini-project: Lesson 11) | Univariate or recursive feature selection is deployed, or features are selected by hand (different combinations of features are attempted, and the performance is documented for each one). Features that are selected are reported and the number of features selected is justified. For an algorithm that supports getting the feature importances (e.g. decision tree) or feature scores (e.g. SelectKBest), those are documented as well. |
| Properly scale features (related mini-project: Lesson 9) | If algorithm calls for scaled features, feature scaling is deployed. |

Pick and Tune an Algorithm

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Pick an algorithm (related mini-project: Lessons 1-3) | At least 2 different algorithms are attempted and their performance is compared, with the more performant one used in the final analysis. |
| Discuss parameter tuning and its importance. | Response addresses what it means to perform parameter tuning and why it is important. |
| Tune the algorithm (related mini-project: Lessons 2, 3, 13) | At least one important parameter tuned with at least 3 settings investigated systematically, or any of the following are true:   * GridSearchCV used for parameter tuning * Several parameters tuned * Parameter tuning incorporated into algorithm selection (i.e. parameters tuned for more than one algorithm, and best algorithm-tune combination selected for final analysis). |

Validate and Evaluate

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Usage of Evaluation Metrics (related mini-project: Lesson 14) | At least two appropriate metrics are used to evaluate algorithm performance (e.g. precision and recall), and the student articulates what those metrics measure in context of the project task. |
| Discuss validation and its importance. | Response addresses what validation is and why it is important. |
| Validation Strategy (related mini-project: Lesson 13) | Performance of the final algorithm selected is assessed by splitting the data into training and testing sets or through the use of cross validation, noting the specific type of validation performed. |
| Algorithm Performance | When tester.py is used to evaluate performance, precision and recall are both at least 0.3. |