PROJECT SPECIFICATION

Identify Fraud from Enron Email

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

CRITERIA	MEETS SPECIFICATIONS
Data Exploration (related lesson: "Datasets and Questions")	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 lesson: "Outliers")	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 lesson: "Feature Selection")	At least one new feature is implemented. Justification for that feature is provided in the written response. The effect of that feature on final algorithm performance is tested or its strength is compared to other features in feature selection. The student is not required to include their new feature in their final feature set.

CRITERIA	MEETS SPECIFICATIONS
Intelligently select features (related lesson: "Feature Selection")	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 lesson: "Feature Scaling")	If algorithm calls for scaled features, feature scaling is deployed.

Pick and Tune an Algorithm

CRITERIA	MEETS SPECIFICATIONS
Pick an algorithm (related lessons: "Naive Bayes" through "Choose Your Own Algorithm")	At least two different algorithms are attempted and their performance is compared, with the best performing one used in the final analysis.

CRITERIA	MEETS SPECIFICATIONS
Discuss parameter tuning and its importance.	Response addresses what it means to perform parameter tuning and why it is important.
Tune the algorithm (related lesson: "Validation")	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 lesson: "Evaluation Metrics")	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.

CRITERIA	MEETS SPECIFICATIONS
Discuss validation and its importance.	Response addresses what validation is and why it is important.
Validation Strategy (related lesson "Validation")	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.