

# Intro to Machine Learning Final Project Rubric

## Overview

This rubric is here to help you understand the expectations for how your project will be evaluated. It is the same rubric that the person evaluating your project will use. You should look at the rubric **before you begin working** on this project **and before you submit it**.

Criteria	Does Not Meet Expectations	Meets expectations
Quality of Code		
Functionality	Code does not perform the functions documented in the writeup, or the writeup does not clearly specify the final analysis strategy.	Code reflects the description in the documentation.
Usability	Dataset, list of features used, or algorithm not exported properly, so that tester.py does not execute.	Dataset, list of features and algorithm are exported using code in poi_id.py, so that it can be checked easily using tester.py
Understanding the Dataset and Question		
Data Exploration (related mini-project: Lesson 5)	Student response does not address characteristics of the dataset, or ignores/gets wrong important characteristics.	Student response addresses the most important characteristics of the dataset and uses these characteristics to inform analysis. Important characteristics include: <ul style="list-style-type: none"><li>total no. of data points</li><li>allocation across classes (POI/non-POI)</li><li>no. of features used</li></ul>
Outlier Investigation (related mini-project: Lesson 7)	Outliers in financial data are not identified, and/or not removed if removal would be appropriate.	Student response identifies outlier(s) in the financial data, and explains how they are removed or otherwise handled.
Optimize Feature Selection/Engineering		
Create new features (related	The only features investigated as possible algorithm inputs are the	At least one new feature implemented. Justification for that feature is

<b>mini-project: Lesson 11)</b>	original features in the dataset provided.	provided in the written response, and the effect of that feature on the final algorithm performance is tested.
<b>Intelligently select features (related mini-project (related mini-project: Lesson 11)</b>	Feature selection is not performed or documented.	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). For an algorithm that supports getting the feature importances (e.g. decision tree), those are documented as well.
<b>Properly scale features (related mini-project: Lesson 9)</b>	Feature scaling is not deployed when called for.	If algorithm calls for scaled features, feature scaling is deployed.
<b>Pick and Tune an Algorithm</b>		
<b>Pick an algorithm (related mini-project: Lessons 1-3)</b>	Only one algorithm attempted.	At least 2 different algorithms attempted, and their performance is compared with the more performant one used in the final analysis.
<b>Tune the algorithm (related mini-project: Lessons 2, 3, 13)</b>	No parameter tuning attempted for an algorithm that has tunable parameters.	At least one important parameter tuned, with at least 3 settings investigated; or any of the following are true: <ul style="list-style-type: none"><li>• GridCV used for parameter tuning</li><li>• Several parameters tuned</li><li>• Parameter tuning incorporated into algorithm selection (i.e. parameters tuned for more than one algorithm, and best algorithm-tune combination selected for final analysis)</li></ul>
<b>Validate and Evaluate</b>		
<b>Usage of Evaluation Metrics (related mini-project: Lesson 14)</b>	Less than 2 quantitative performance metrics deployed, or student cannot articulate what those metrics measure.	Precision and recall are used to evaluate performance performance, and student articulates what those metrics measure.
<b>Validation Strategy (related mini-project: Lesson 13)</b>	There is no data held out for testing.	The data is split into training and testing sets, with the testing data used for assessing overall analysis performance; or k-fold cross validation is deployed
<b>Analysis Performance</b>	Precision and recall both consistently below 0.3	Precision and recall above 0.3

## Before you Submit

1. After you've completed all the auto-graded questions and answered the written questions, go through each rubric item and do your best to honestly evaluate where you think your project falls.
2. If you think your project "does not meet expectations" for **any** criteria item, you should make any necessary changes.
3. Once you're confident that your project "meets expectations" or "exceeds expectations," you can submit by emailing your written responses to [introml-project@udacity.com](mailto:introml-project@udacity.com)

## How Grading Works

1. Your project evaluator will be able to see all your code submissions. They will use this rubric to evaluate your code as well as your written responses.
2. Your grade will simply be "pass, meets expectations," "pass, exceeds expectations," or "doesn't pass,"
  - a. You earn "pass, meets expectations" if **all** criteria "meet expectations."
  - b. You earn "pass, exceeds expectations," if all criteria "exceed expectations" (when possible).
  - c. Your project "doesn't pass" if **any** criteria are graded as "doesn't meet expectations." In this case, you will have the opportunity to revise and resubmit.