Intro to Machine Learning Final Project Rubric

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	Meets expectations
Quality of Code	
Functionality	Code reflects the description in the answers to questions in the writeup. The write up 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	
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 ● 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. Outliers are removed or retained as appropriate.
Optimize Feature Selection/Engineering	
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.

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	
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.
Tune the algorithm (related mini-project: Lessons 2, 3, 13)	Response addresses what it means to perform parameter tuning and why it is important. 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	
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.
Validation Strategy (related mini-project: Lesson 13)	Response addresses what validation is and why it is important. 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.

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," if you are a Data Analyst Nanodegree student, go

to your Udacity home and submit it for review. If you are a free student, share with your networks and get feedback.

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 do not meet expectations. In this case, you will have the opportunity to revise and resubmit.

Published by Google Drive – Report Abuse – Updated automatically every 5 minutes