|  |  |
| --- | --- |
| Criteria | Meets expectations |
| **Quality of Code** | |
| Functionality | Code reflects the description in the documentation. |
| Usability | Dataset, list of features and algorithm are exported using code in poi\_id.py, so that it can be checked easily using tester.py  The function in the tester.py will delete two more data, so the result is slightly different with the poi\_id.py.  However, it would be unreasonable to push the transformed features from PCA into the dataset. (It’s hard to name the transformed features or the components to each person) For the pca part, the stratified k fold validation is written into the code. The code file is named poi\_id.py |
| **Understanding the Dataset and Question** | |
| Data Exploration (related mini-project: Lesson 5) | * total no. of data points * allocation across classes (POI/non-POI) * no. of features used   These three points have been addressed. The no. of features used went through two steps of feature selection. (Available data points and select k best) |
| Outlier Investigation (related mini-project: Lesson 7) | Two outliers have been removed from the data. “TOTAL” and “THE TRAVEL AGENCY IN THE PARK” which do not represent any people. |
| **Optimize Feature Selection/Engineering** | |
| Create new features (related mini-project: Lesson 11) | Another variable has added in the learning algorithm which is the most weighted variable calculated by the select k best method. |
| Intelligently select features (related mini-project (related mini-project: Lesson 11) | Two steps of feature selection: 1) Available data points and 2) select k best method. Also the PCA has been applied to test.  The select k best method is based on ANOVA test and the F value and p value have been reported. |
| Properly scale features (related mini-project: Lesson 9) | Feature scaling which rescale the features into the range from 0 to 1 has been used. |
| **Pick and Tune an Algorithm** | |
| Pick an algorithm  (related mini-project: Lessons 1-3) | AdaBoost and logistic regression |
| Tune the algorithm (related mini-project: Lessons 2, 3, 13) | GridSearchCV function has been used to tune different parameters. However, in the poi\_id.py, I commented the line with several tuned parameters, because it will not give you the best precision and recall value that the GridSearchCV function chooses. The reason could be that the GridSearchCV function uses the accuracy to choose the best parameters. However, in this unbalance dataset, the best accuracy might not equal to best precision and accuracy  The base estimators include decision tree classifier, SVC, random forest classifier and none. |
| **Validate and Evaluate** | |
| Usage of Evaluation Metrics (related mini-project: Lesson 14) | The precision and recall are documented in the final report. |
| Validation Strategy (related mini-project: Lesson 13) | The stratified k fold cross validation has been integrated into the poi\_id.py and poi\_id\_pca.py to evaluate the performance. |
| Analysis Performance | Precision = 0.6095  Recall = 0.3889 |