**Does Not Meet Specifications**

**Quality of Code**

SPECIFICATION

Code reflects the description in the documentation i.e. code performs the functions documented in the writeup and the writeup clearly specifies the final analysis strategy.

 MEETS SPECIFICATION

**Reviewer Comments**

Your submission is quite good and concise.

Ideally every step/choice of parameters and algorithm selection, tuning and validation should be followed by a table with the adequate figures reporting the impact of that specific step/choice on the chosen performance metrics and a consequent rationale for the choices made in the light of those results.  
This happens sometimes in your report but not always: Doing so would help the reader properly understanding the consequent decision process. It is very important, in every scientific work, to explain and justify every step/choice as thoroughly as possible, no matter how obvious to you the various steps might look.

There is as well a project rubric that mandatory requires some definitions to be given and the importance of some factors to be discussed.

I will, section by section, point out the areas where more information is required, please invest some more time on crafting the best possible project for you to showcase in your portfolio or to your prospect employer.

That said your actual results are quite good, well done!.  
Keep up your good work!

SPECIFICATION

Dataset, list of features and algorithm are exported using code in poi\_id.py, so that it can be checked easily using tester.py.

 MEETS SPECIFICATION

**Reviewer Comments**

Awesome Job!

**Understanding the Dataset and Question**

SPECIFICATION

Student response addresses the most important characteristics of the dataset and uses these characteristics to inform analysis. Important characteristics include:

* total number of data points
* allocation across classes (POI/non-POI)
* number of features used

 MEETS SPECIFICATION

**Reviewer Comments**

Awesome Job!. Good description provided.

Please note this dataset characteristics have some impacts over the strategy to follow in this problem:

1. The small data set is why the tester.py file uses StratifiedShuffleSplit instead of a simpler cross-validation method such as TrainTestSplit. StratifiedShuffleSplit will make randomly chosen training and test sets multiple times and average the results over all the tests.
2. The data is unbalanced with many more non-POIs than POIs. StratifidShuffleSplit also makes sure that the ratio of non-POI:POI is the same in the training and test sets as it was in the larger data set. The unbalanced data is also why we use precision and recall instead of accuracy as our evaluation metric.

SPECIFICATION

Student response identifies outlier(s) in the financial data, and explains how they are removed or otherwise handled.

 MEETS SPECIFICATION

**Reviewer Comments**

Awesome Job!. TOTAL is definitely and outlier.

**Optimize Feature Selection/Engineering**

SPECIFICATION

At least one new feature implemented. Justification for that feature is provided in the written response, and the effect of that feature on the final algorithm performance is tested.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

Good work engineering your new features, however you need to include also your motivations to create them and evaluate the impact of your created features over your final classifier.

STEPS TO PASS THIS SECTION:

1. Include in your written response your reasons for these features.
2. Include in your written response a table showing precision&recall values of your final classifier with and without using your new features.

SPECIFICATION

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.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

Although I see in poi\_id.py your final feature list is: ['poi','incsum', 'incema'] , I can't find any information in your written response about the feature selection process followed to generate this final list of features.

Feature selection process is key in Machine Learning problems, since it defines the set of features that serves as input for the final algorithm, for this reason this process must follow an **exhaustive, robust and scientific methodology, without any room for intuition**. Ideally each feature set being tested is evaluated in terms of precision&recall scores, the set with higher values is the final set of features to use. Of course this process must be well documented in your written response, for example you can include a table for your different set of features being tested and its corresponding precision&recall values. This information would allow us to understand which set of features maximizes algorithm's performance.

Also, it is very important that your methodology is exhaustive, generating sets of features randomly or using intuition is not part of the requested methodology. Scikit includes [tools](http://scikit-learn.org/stable/modules/feature_selection.html#feature-selection)(like SelectKBest) that generates different sets of features based on score functions, for example in SelectKBest this is done by changing the value of the "K" parameter.

Finally, different algorithms may have different set of features, but also note that same algorithms may present a different set of features if tuned differently. So please define in your written response which is the classifier used during the process.

STEPS TO PASS THIS SECTION:

1. Include some comments in your written response on the classifier used in this process.
2. Include a table in your written response a table with the precision&recall values of the different sets being tested.(You can generate sets by playing with K in SelectKBest). Your final set of features is the one that maximizes precision&recall for your algorithm.

SPECIFICATION

If algorithm calls for scaled features, feature scaling is deployed.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

Some algorithms tested (KNN, KMEANS, SVC) call for scaling since are based on Euclidean distances. I can't find any reference in your written response, neither in poi\_id.py of scaling being deployed for these algorithms.

For example, for DecissionTrees If you think about it, the decision is like: "is feature x\_i >= some\_val?" Here, it doesn't matter on which scale this feature is.

Algorithms where feature scaling matters are:

* k-means if you use, for example, Euclidean distance since you typically want all features to contribute equally
* k-nearest neighbors (see k-means)
* logistic regression, SVMs, perceptrons, neural networks etc

There are more that I am not going to list here (above are the most prominent examples in my opinion). Actually, I always recommend you to think about the algorithm and what its doing, and then it typically becomes obvious whether you want to scale your features or not. Please review the lecture videos, those are good resources.

STEPS TO PASS THIS SECTION:

1. Deploy scaling when algorithms call for it.

**Pick and Tune an Algorithm**

SPECIFICATION

At least 2 different algorithms attempted, and their performance is compared with the more performant one used in the final analysis.

 MEETS SPECIFICATION

**Reviewer Comments**

Awesome Job!. You tested a variety of algorithms and included their performance, and included some comments in your written response on what is the algorithm that best performs in terms of precision&recall.

SPECIFICATION

Response addresses what it means to perform parameter tuning and why it is important.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

Please include some comments in your written response of what is tuning and why it is important. Lecture videos are a good resource, but also [this link](http://machinelearningmastery.com/how-to-improve-machine-learning-results/).

SPECIFICATION

At least one important parameter tuned, with at least 3 settings investigated; or any of the following are true:

* GridCV 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).

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

Although you tested some algorithms with a different parameters setup, for example:

RandomForestClassifier(n\_estimators=100)

RandomForestClassifier(max\_depth=5, n\_estimators=10, max\_features=1)

RandomForestClassifier(max\_depth=10, n\_estimators=15, max\_features=2)

A tuning process include the evaluation of a particular algorithm for each set of parameters (you do this part) and a discussion process where it is evaluated, in terms of precision and recall, which is the best set of parameters (this part is not included in your report).

STEPS TO PASS THIS SECTION:

1. For each algorithm tuned, please investigate 1 parameter at least, with at least 3 settings
2. Include a table showing precision&recall values for each configuration attempted
3. Include some comments on what is the most suitable parameter's configuration

**---------------------**

As a recommendation, I really encourage you to become familiar with [SearchGridCV](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html" \t "_blank), with using it to tune your algorithm you can:

1. Do a more robust search over your different parameters to find the best parameters.
2. Use the "scoring" parameter to lead the parameter search process to maximize any parameter of your choice, for example precision, recall...
3. Using the "cv" parameter you can pass a cross validation object to validate your search results.

**Validate and Evaluate**

SPECIFICATION

Precision and recall are used to evaluate performance performance, and student articulates what those metrics measure.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

Please include some comments of what is precision and recall in terms of POI detection. Lecture videos are good resources, but also [this link](https://en.wikipedia.org/wiki/Precision_and_recall)

SPECIFICATION

Response addresses what validation is and why it is important.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

You need to include some comments of what is validation and why is it important. Lecture videos are good resources, but also [this link](http://scikit-learn.org/stable/modules/cross_validation.html).

SPECIFICATION

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.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

As described in the code review section, test\_classifier is provided to you as a reference and should not be part of your code. This means you need to deploy your own validation method.

Also note KMEANS is designed for unsupervised clustering problems, this one is a [**Supervised Classification**](https://en.wikipedia.org/wiki/Statistical_classification) problem, so KMEANS is not designed for this kind of problems.

STEPS TO PASS THIS SECTION:

1. Don't use KMEANS in this problem, it is designed for clustering for unsupervised learning.
2. Deploy your own validation process.

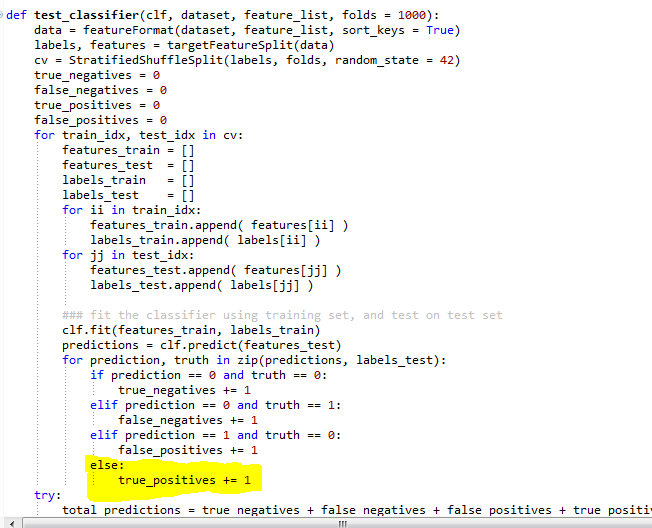
**Why you get good precision&recall values using test\_classifier()?**

Note KMEANS is gathering features into groups, for example if you use:

KMeans(n\_clusters=5, n\_init=1, init='random')

Your algorithm outputs is any value in the range [0,1,2,3,4], since kmeans is configured for 5 different clusters.

Since test\_classifier() expects your classifier output to be in the range [0,1], when performing the evaluation if classifier's output is higher than 1, there is a bug in test\_classifier() that makes it assume it is a "true\_positive" when it is actually not.

[](https://udacity-github-sync-content.s3.amazonaws.com/_imgs/2613/1439892830/Capture.PNG)

SPECIFICATION

Precision and recall are both at least 0.3.

 DOES NOT MEET SPECIFICATION

**Reviewer Comments**

As mentioned above, Kmeans is not a suitable algorithm in this problem and test\_classifier is not designed to validate its performance. So, we can't evaluate your algorithm.

**---------**

This project is a good start, please follow the instructions provided along this review and don't hesitate to contact us or consult our forums if you have questions. Good luck with your next submission!