UDACITY

PROJECT

Identify Fraud from Enron Email

A part of the Data Analyst Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Great job improving your project based on feedback!

Your project shows the hard work you put in. I did want to say that it is an outstanding work and you should be proud. The things that make it stand out is the structure, graphs!, and references. Although it may seem like small details, they make a huge difference for a reader.

Keep up the awesome work!

Quality of Code

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.

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

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.**

Good summary of all the most importan features. The only suggestion I can make here is that the answer to section 2 could make more emphasis on the unbalanced allocation across classes. This is arguably the most important characteristic of the data set given that it defines what metrics will be good to use and what validation methods are better.

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

All three of the outliers were removed.

Optimize Feature Selection/Engineering

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.

Two new features: `fraction_from_poi` and `fraction_to_poi` and their effects reported :)

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.

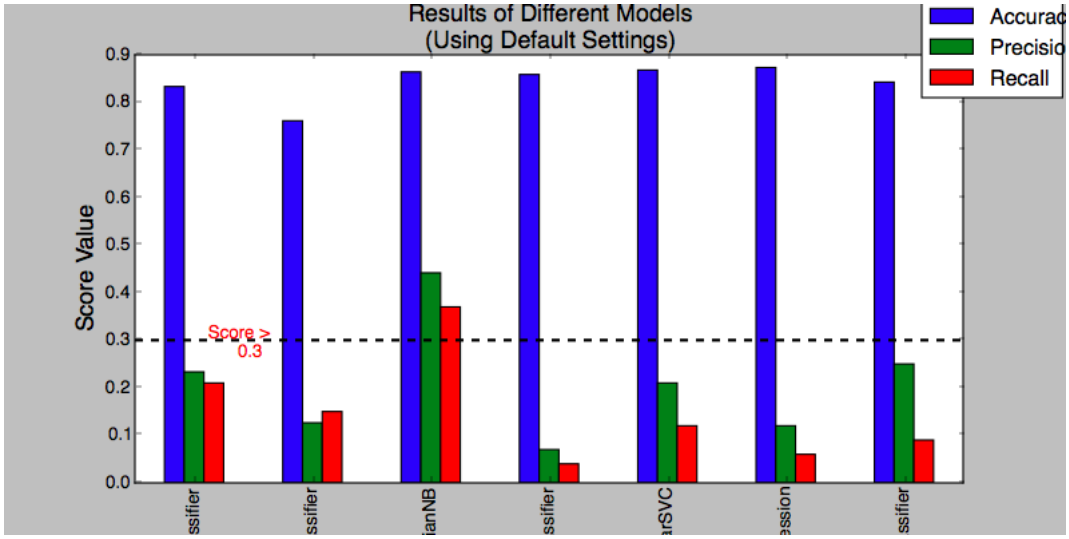
`SelectKBest` and *Recursive Feature Elimination* were used to select top performing features.

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

Pick and Tune an Algorithm

At least two different algorithms are attempted and their performance is compared, with the best performing one used in the final analysis.

Great job plotting the results of the algorithms!



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).**

Great job using `GridSearchCV`

Validate and Evaluate

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.

Both precision and recall were used to evaluate and defined correctly in the context of the project.

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.

When tester.py is used to evaluate performance, precision and recall are both at least 0.3.

```
GaussianNB(priors=None)
    Accuracy: 0.86647    Precision: 0.49905    Recall: 0.39600    F1: 0.44159    F2: 0.41306
    Total predictions: 15000    True positives:  792    False positives:  795    False negatives: 1208    True negatives: 12
20
```

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ