UDACITY

PROJECT

Identify Fraud from Enron Email

A part of the Data Analyst Nanodegree Program

PROJECT REVIEW

CODE REVIEW 3

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

Dear student,
First of all, thank you for your submission! It's a very solid one!
You did a very good job in this submission, but we still have a couple of issues to address before going ahead.
Please refer to the comments below for details on the required changes and suggestions to further improve certain aspects of the report and code.
I really hope you find them useful!

Keep up the great work! We are looking forward to your next submission! 😀

Have a great 2018!! 🎉

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.**

AWESOME

I really appreciate the effort put on this section. It presents really important characteristics of the dataset!

SUGGESTION

A detail that deserves a bigger attention is the ratio between POIs and non-POIs in our dataset. As it was correctly stated in the report, we have only 18 POIs out of 146 individuals! This will be especially problematic when creating classification models due to a phenomenon called class imbalance. This is really important since it affects how learning, evaluation and validation should happen!

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

SUGGESTION

In addition to `TOTAL`, we have another two outliers which are pretty clear and easy to find:

- One is not a real person (take a look at the names and you will find it!)
- The second has all of its values missing!

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.

REQUIRED

Very solid start to this section! Five new features were implemented and used as part of the feature selection procedure.
Unfortunately, I need to mark this as requiring changes since the rubric asks us to provide the impact of the new features.
We have two options to tackle this:

1. Present the features' scores for the original feature set and then compare them with the scores obtained by the dataset with the new features. To meet this criterion, just a small paragraph discussing how good or bad new feature is enough.
2. Or train a classifier in the original subset of features and another with the dataset with the newly generated ones. Finally, we could assess their results and determine the impact of the new features.

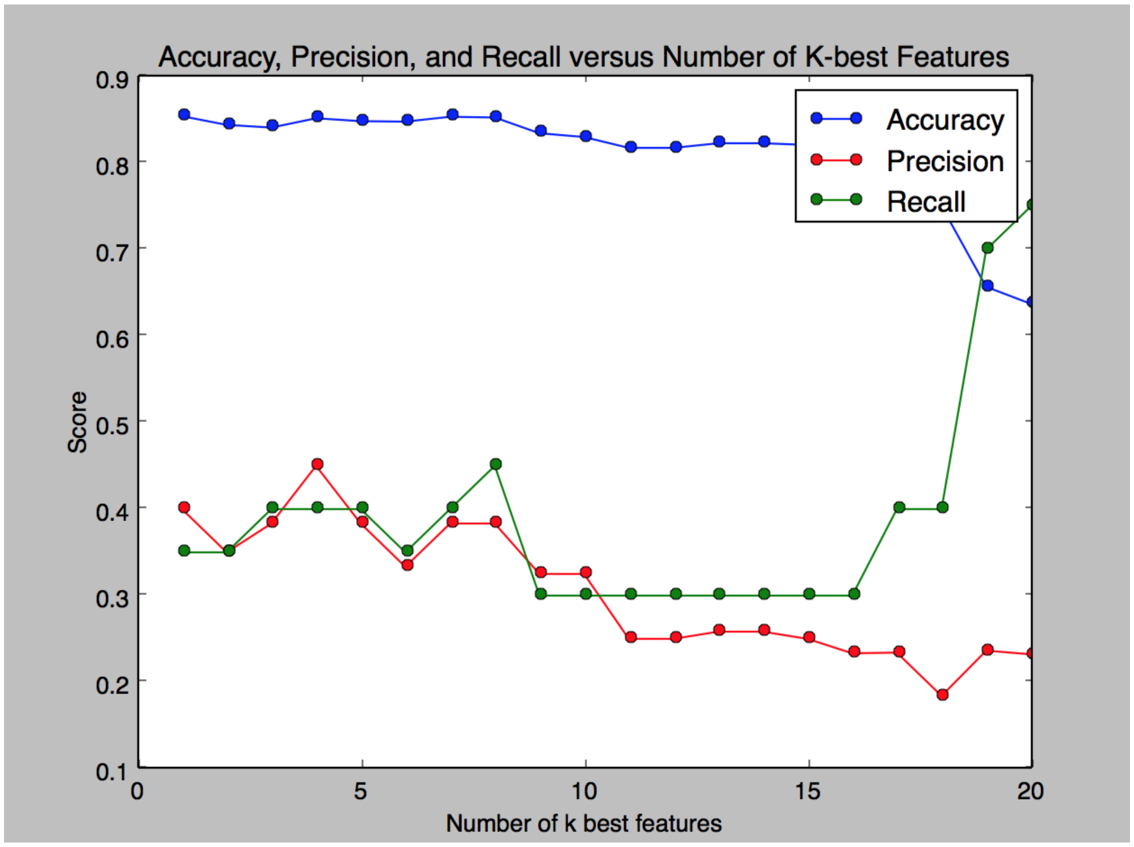
Univariate or recursive feature selection is deployed, or features are selected by hand (different combinations of features are attempted, and the performance is documented for each one). Features that are selected are reported and the number of features selected is justified. For an algorithm that supports getting the feature importances (e.g. decision tree) or feature scores (e.g. SelectKBest), those are documented as well.

REQUIRED

Overall, we have a solid start in this section! Good job!
To meet this criterion, we only need to clarify why the top `k=5` features have been selected and not another number.
For instance, why not `k=2` or `k=20`?
To meet this criterion, we need to make sure we detail how `k` has been selected and why.
I'm not sure how `k` has been selected thus far, but I will leave two suggestions below.

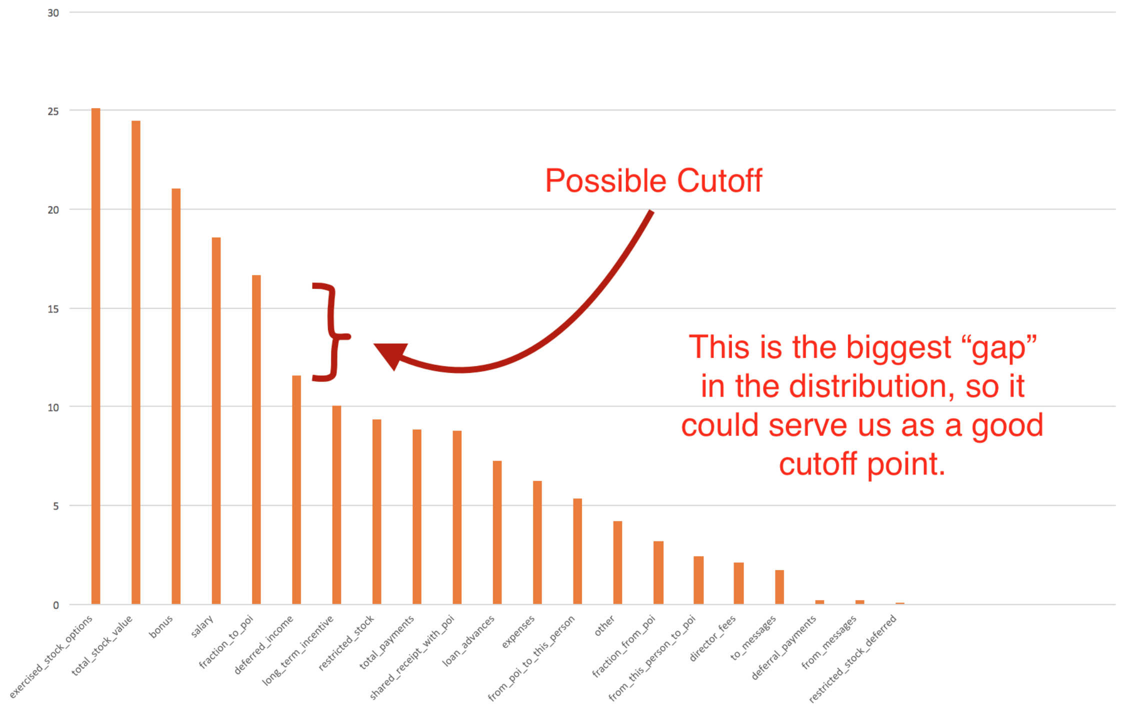
GUIDANCE

Regarding the determination of which value of `k` should be used, my suggestion is to test several values and track the performances obtained!
For instance, we could create a plot as the one presented below, where precision and recall are presented for different values of `k`. This way, we can clearly determine a fair value of `k` that presents a fair precision while also depicting a recall above our requested threshold of 0.3!



ANOTHER POSSIBILITY

Another possibility would be to try to find a good "cut-off" point in our scores' distribution. We could create a plot like the one below and find the "gap" in the distribution, where the scores start to decrease more rapidly. This would allow us to find a good cut-off point that splits our feature set into the "best" features and the "not so good" 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.

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

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.

REQUIRED

In the report, we have the following:

> Based on the above definition, we can understand that the accuracy shows the number of poi that was accurately predicted by the model, while the precision refers to how well the model didn't misclassify someone as a poi and recall shows how the model correctly predicted a poi as a poi.

To meet this criterion, we need to revisit and clarify each of the evaluation metrics used.
Let me see if I can help with a few guiding questions:

- Regarding accuracy: accuracy measures the ratio of predictions of our classifier that are correct, regardless of whether they are POI or not!
- Thinking about precision: are the POIs flagged by our classifier really POIs?
- Now about recall: is our classifier able to identify all the POIs it should?

Finally, I believe the first answer provided in this page could also help!

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.

SUGGESTION

Even though the report mentions that `StratifiedShuffleSplit` has been used, it would be nice to further detail one of its most important traits: **stratification.**
In this project, we are dealing with a small and imbalanced dataset.

As seen throughout the lectures, working with smaller datasets is hard and in order to make validation models robust, we often go with **k-fold cross-validation**, which is what we do when we use a shuffle split.
But well, in this project (and so many others we have in our day to day work), a stratified shuffle split is of choice. When dealing with small imbalanced datasets, it is possible that some folds contain almost none (or even none!) instances of the minority class. The idea behind stratification is to keep the percentage of the target class as close as possible to the one we have in the complete dataset. Please take a look here and here for more.

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

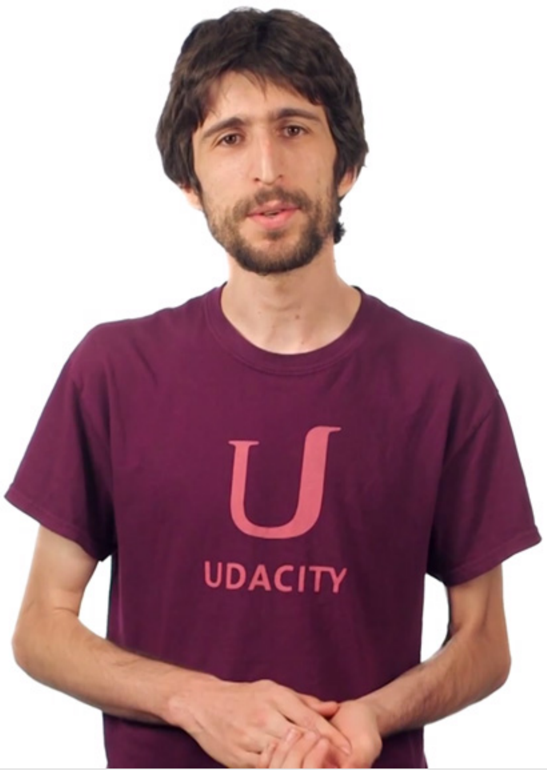
OUTPUT

```
GaussianNB(priors=None)
    Accuracy: 0.90409     Precision: 0.46055     Recall: 0.32100     F1: 0.37831     F2: 0.34171
    Total predictions: 11000     True positives:  321     False positives:  376     False negatives:  679     True neg
atives: 9624
```

RESUBMIT

DOWNLOAD PROJECT

3 CODE REVIEW COMMENTS ›



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH

Rate this review

Student FAQ