UDACITY

Logout

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Dear student,

well done improving your submission and addressing the last remaining issue, you made it for an excellent project. There's not much I can actually further add to your good work aside from a couple of Pro Tips regarding stratification and feature scaling with support vector machines. If you are enthusiastic about process optimization though, when it comes to machine learning, I left a Pro Tip for you in the code section. Please be advised that it is quite advanced, therefore don't worry if you don't get it immediately, you could keep it and use it later on as you progress in the Nanodegree.

Congratulations on passing your exam!

Classification vs Regression

Student is able to correctly identify which type of prediction problem is required and provided reasonable justification.

Exploring the Data

Student response addresses the most important characteristics of the dataset and uses these characteristics to inform their decision making. Important characteristics must include:

- **Number of data points**
- **Number of features**
- **Number of graduates**
- **Number of non-graduates**
- **Graduation rate**

Preparing the Data

Code has been executed in the iPython notebook, with proper output and no errors.

Training and test sets have been generated by randomly sampling the overall dataset.

Pro Tip: Data assessment:
When dealing with the new data set it is good practice to assess its specific characteristics and implement the cross validation technique tailored on those very characteristics, in our case there are two main elements:

1. Our dataset is **small**.
2. Our dataset is slightly **unbalanced**. (There are more passing students than on passing students)

We could take advantage of K-fold cross validation to exploit small data sets. Even though in this

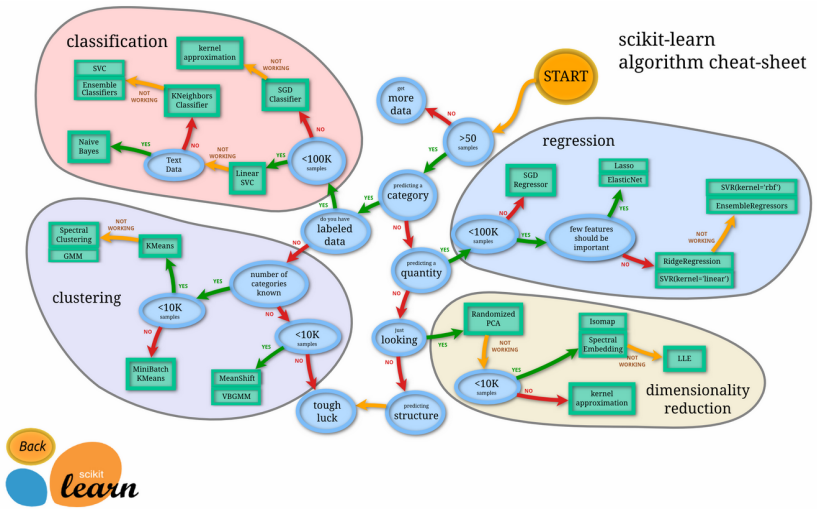
case it might not be necessary, should we have to deal with heavily unbalance datasets, we could address the unbalanced nature of our data set using Stratified K-Fold and Stratified Shuffle Split Cross validation, as stratification is preserving the preserving the percentage of samples for each class.
http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html
http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedKFold.html
As for the initial train test split you can obtain stratification by simply using **stratify = y_all**:

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_all, y_all, stratify
= y_all, test_size=95, random_state=42)
```

Training and Evaluating Models

Three supervised models are chosen with reasonable justification. Pros and cons for the use of each model are provided, along with discussion of general applications for each model.

Pro Tip:
When choosing which algorithm to use this interactive map provides a good starting point:
http://scikit-learn.org/stable/tutorial/machine_learning_map/



All the required time and F1 scores for each model and training set sizes are provided within the chart given. The performance metrics are reasonable relative to other models measured.

Optional: You could use a loop to go through the algorithms:

```
for clf in [clf_A, clf_B, clf_C]:
    print "\n{}: \n".format(clf.__class__.__name__)
    for n in [100, 200, 300]:
        train_predict(clf, X_train[:n], y_train[:n], X_test, y_test)
```

Choosing the Best Model

Justification is provided for which model seems to be the best by comparing the computational cost and accuracy of each model.

Student is able to clearly and concisely describe how the optimal model works in laymen

terms to someone what is not familiar with machine learning nor has a technical background.

The final model chosen is correctly tuned using gridsearch with at least one parameter using at least three settings. If the model does not need any parameter tuning it is explicitly stated with reasonable justification.

Optional: It is not mandatory to scale features for this project, please note though that feature scaling should be implemented with SVMs, performance might be affected otherwise:
http://stats.stackexchange.com/questions/65094/why-scaling-is-important-for-the-linear-svm-classification
http://scikit-learn.org/stable/modules/preprocessing.html

The F1 score is provided from the tuned model and performs approximately as well or better than the default model chosen.

Quality of Code

Code reflects the description in the documentation.

Pro Tip (Advanced): You could actually go well beyond grid search and implement 'pipelines' where the whole machine learning process becomes 'grid-searchable' and you can parameterize and search the whole process though cross validation.
http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html
And yes you can try out several algorithms automatically as well too! Watch out though this is pretty advanced stuff, here is a great, informative, top notch tutorial from Zac Sewart!
http://zacstewart.com/2014/08/05/pipelines-of-featureunions-of-pipelines.html

DOWNLOAD PROJECT

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Student FAQ