DTSC670: Foundations of Machine Learning Models

Module 2

Assignment 5 Fall 2 2022-Jung-Ping Wu

Scikit-learn is an incredibly popular machine learning library. Written in Python, it is designed to be simple and easy to use. There are 5 key design principles of the Scikit-Learn library:

1. Consistency
2. Inspection
3. Nonproliferation of classes
4. Composition
5. Sensible defaults

The first is consistency: It has a basic command API for all the objects. It also adopts simple conventions and limits to minimize the number of methods. The main objects include:

1. Estimator. The most common one. Implements a “fit” method to learn from data:

estimator = estimator.fit(data, targets)

estimator = estimator.fit(data)

One of the nice features is when you construct an estimator, it passes in hyper-parameters, but does not pass in the training data; the training data is passed in via the fit method. All of the estimator’s hyperparameters are accessible directly via public instance variables. Estimators can be used with any of the algorithms like classification, regression, clustering or even with a transformer that extracts useful features from raw data.

2. Transformers: This is for filtering or modifying the data (in a supervised or unsupervised way), which implements the following behavior:

new\_data = transformer.transform(data)

Or, to fit and transform together:

new\_data = transformer.fit\_transform(data)

We can clean, reduce, expand or generate feature representations easily with a library of transformers.

3. Predictor: Extends the Estimator interface. In order to make the model work, an estimator has to implement a predict method. After a model is trained, the predictor returns predicted labels for a given input feature vector. It also provides probabilities as well as prediction scores. This is another design principle.

prediction = predictor.predict(data)

-For supervised learning, or some unsupervised problems)

probability = predictor.predict\_proba(data)

-Classification algorithms

The predict method that a model implements as part of the Predictor interface does the work of predicting results.

The second design principle is inspection: All an estimator’s hyperparameters and learned parameters are accessible via public instance variables. One of the main goals of machine learning is predictive performance. The inspection of a model can be a powerful tool to help us fit better models.

The third design principle is non-proliferation of classes : Datasets are represented as NUMPY or SciPy sparse matrices. Hyperparameters are presented as normal Python strings.

The fourth design principle is Composition. SciKit learn objects are easy to arrange and compose to create different transformations and estimations.

The last principle is Sensible Defaults. Providing sensible defaults for operations makes for an efficient way (for the programmer) to produce a base-line solution for a task. It also helps to ensure that an operation is performed in a sensible way.

Scikit-Learn is one of the most practical machine learning libraries for Python. Due to the variety of features, it can be used for data analysis and predictions. It is friendly to use, even for the beginners (plus it is well-documented). It is useful for advanced practitioners: you may use only a few lines of code and have a complete solution. Overall it is flexible and combines well with other Python libraries, for example: [matplotlib for plotting](https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/), [numpy for array vectorization](https://www.activestate.com/resources/quick-reads/how-to-build-a-numpy-array/), and [pandas for dataframes](https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/).