Machine Learning

|SCIKIT-LEARN DESIGN

Prof. Dr. Mostafa Elhosseini

https://youtube.com/drmelhosseini

Agenda

- SCIKIT-LEARN DESIGN

Estimators

- In essence, it is a **Python object** that implements the methods **.fit(X, y)** and **.transform(X)**, with X being the input data and y being the target variables
- All **objects** share a consistent and simple interface
- The main methods are **.fit()**, which is used for **learning from the data**, and **.predict() or .transform()**, which are used for **applying the learned** transformation/model to new data.
- Any object that can **estimate some parameters based on a dataset** is called an estimator

Estimators

- **SimpleImputer** is an estimator.
- The **estimation itself is performed by the fit() method**, and it takes a **dataset** as a parameter, or two for supervised learning algorithms—the second dataset contains the **labels**.
- Any other parameter needed to guide the estimation process is considered a **hyperparameter** (such as a **SimpleImputer's strategy**), and it must be set as an instance variable (generally via a constructor parameter).

Estimators

Types of Estimators:

- **Classifier**: Estimators used for classification tasks, predicting discrete labels for given input data.
- **Regressor**: Estimators used for regression tasks, predicting continuous values.
- **Transformer**: Estimators that are used to **transform input data before feeding** it into a model, such as scaling, normalization, or feature extraction methods.
- **Model Selection**: Tools for comparing, validating, and choosing parameters and models, leading to improved model performance.

Transformers

- Estimators that are used to transform input data before feeding it into a model, such as scaling, normalization, or feature extraction methods.
- Some estimators (such as a **SimpleImputer**) can also transform a dataset; these are called **transformers**.
- All transformers also have a convenience method called **fit_transform()**, which is equivalent to calling fit() and then transform() (but sometimes fit_transform() is optimized and runs much faster).

Predictors

- Some estimators, given a dataset, are capable of making predictions; they are called **predictors**.
- **Classifier**: Estimators used for classification tasks, predicting discrete labels for given input data.
- **Regressor**: Estimators used for regression tasks, predicting continuous values.

Predictors

- LinearRegression model is a predictor: given a country's GDP per capita, it predicted life satisfaction.
- A predictor has a **predict()** method that takes a dataset of new instances and returns a dataset of corresponding predictions. It also has a **score()** method that measures the quality of the predictions, given a test set (and the corresponding labels, in the case of supervised learning algorithms).

Inspection

- All the **estimator's hyperparameters** are accessible directly via public instance variables (e.g., **imputer.strategy**), and all the estimator's learned parameters are accessible via public instance variables with an underscore suffix (e.g., **imputer.statistics_**).

Nonproliferation of classes

- Datasets are represented as NumPy arrays or SciPy sparse matrices, instead of homemade classes. Hyperparameters are just regular Python strings or numbers.

Composition

- Existing building blocks are reused as much as possible. For example, it is easy to create a Pipeline estimator from an arbitrary sequence of transformers followed by a final estimator.

Sensible defaults

- Scikit-Learn provides reasonable default values for most parameters, making it easy to quickly create a baseline working system.

Note

- As with all estimators, it is important to **fit the scalers to the training data only**: never use fit() or fit_transform() for anything else than the training set.
- Once you have a trained scaler, you can then use it to **transform() any other set**, including the validation set, the test set, and new data.
- Note that while the training set values will always be scaled to the specified range, if new data contains **outliers**, these may end up scaled outside the range. If you want to avoid this, just **set the clip hyperparameter to True**.