Methods for loading other datasets:

- Boston house-prices dataset – **datasets.load_boston()**
- Diabetes dataset – **datasets.load_diabetes()**
- Wisconsin breast cancer dataset – **datasets.load_breast_cancer()**
- Wine dataset – **datasets.load_wine()**

Splitting the Dataset into Training and Test Sets

A core practice in machine learning is to split the dataset into different partitions for training and testing. Scikit-learn has a convenient method to assist in that process called **train_test_split(X, y, test_size=0.25)**, where **X** is the design matrix or dataset of predictors and **y** is the target variable. The split size is controlled using the attribute **test_size**. By default, test_size is set to 25% of the dataset size. It is standard practice to shuffle the dataset before splitting by setting the attribute **shuffle=True**.

```
# import module
from sklearn.model_selection import train_test_split
# split in train and test sets
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target,
shuffle=True)
```

```
X_train.shape
'Output': (112, 4)
X_test.shape
'Output': (38, 4)
y_train.shape
'Output': (112,)
y_test.shape
'Output': (38,)
```

Preprocessing the Data for Model Fitting

Before a dataset is trained or fitted with a machine learning model, it necessarily undergoes some vital transformations. These transformations have a huge effect on the performance of the learning model. Transformations in Scikit-learn have a **fit()** and **transform()** method, or a **fit_transform()** method.