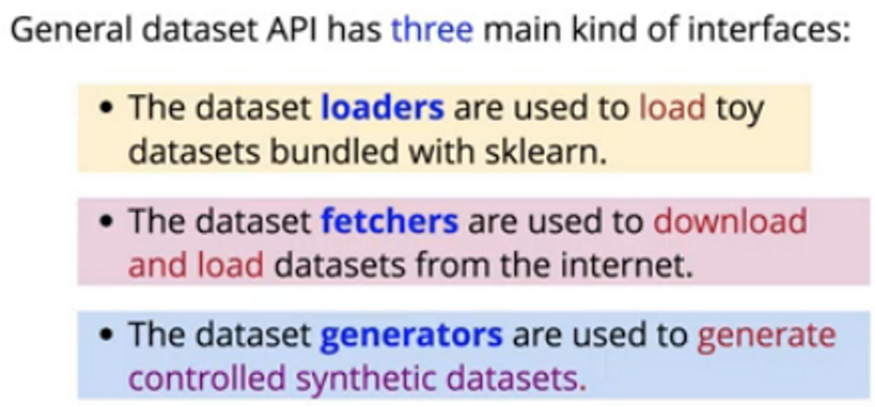
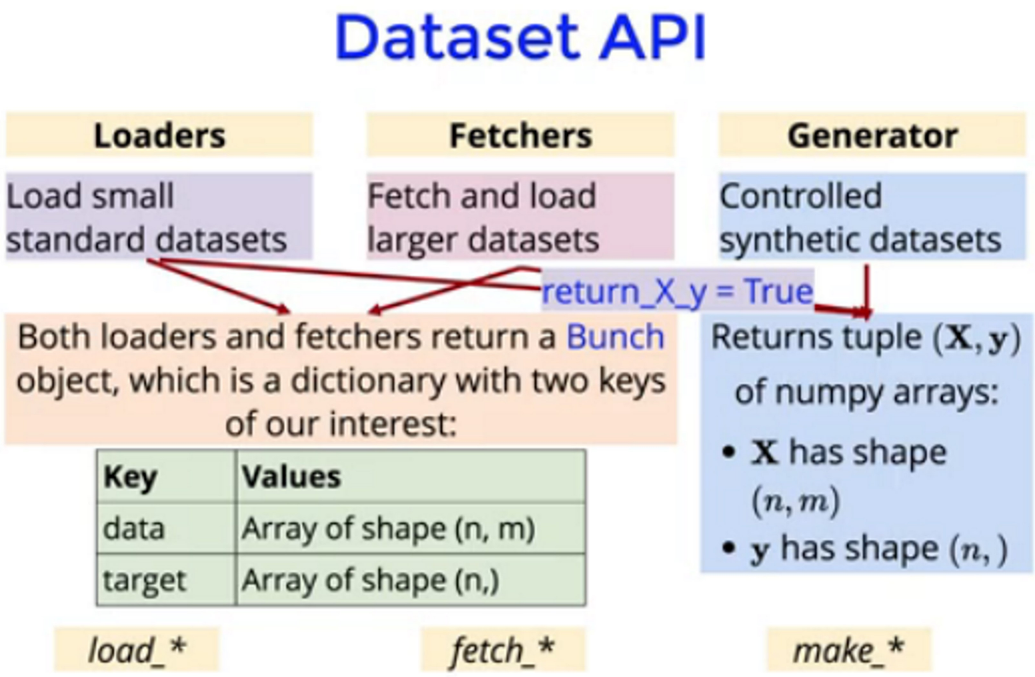
Data Loading

29 January 2023 05:36 PM



General dataset API has three main kind of interfaces:

- The dataset **loaders** are used to load toy datasets bundled with sklearn.
- The dataset **fetchers** are used to download and load datasets from the internet.
- The dataset **generators** are used to generate controlled synthetic datasets.



Dataset API

Loaders

Load small standard datasets

Fetchers

Fetch and load larger datasets

Generator

Controlled synthetic datasets

return_X_y = True

Both loaders and fetchers return a Bunch object, which is a dictionary with two keys of our interest:

Key	**Values**
data	Array of shape (n, m)
target	Array of shape (n,)

Returns tuple $(\mathbf{X}, \mathbf{y})$ of numpy arrays:

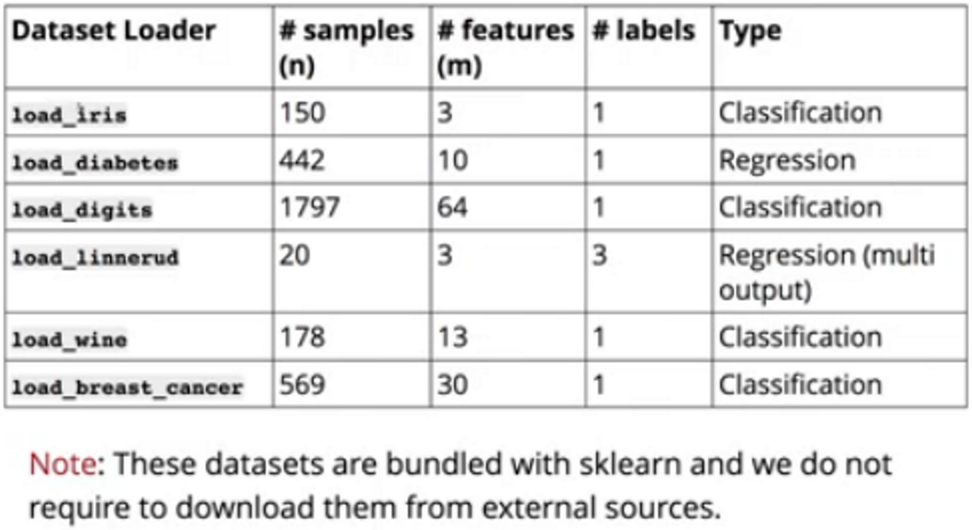
- $\mathbf{X}$ has shape (n, m)
- $\mathbf{y}$ has shape $(n,)$

*load_**

*fetch_**

*make_**

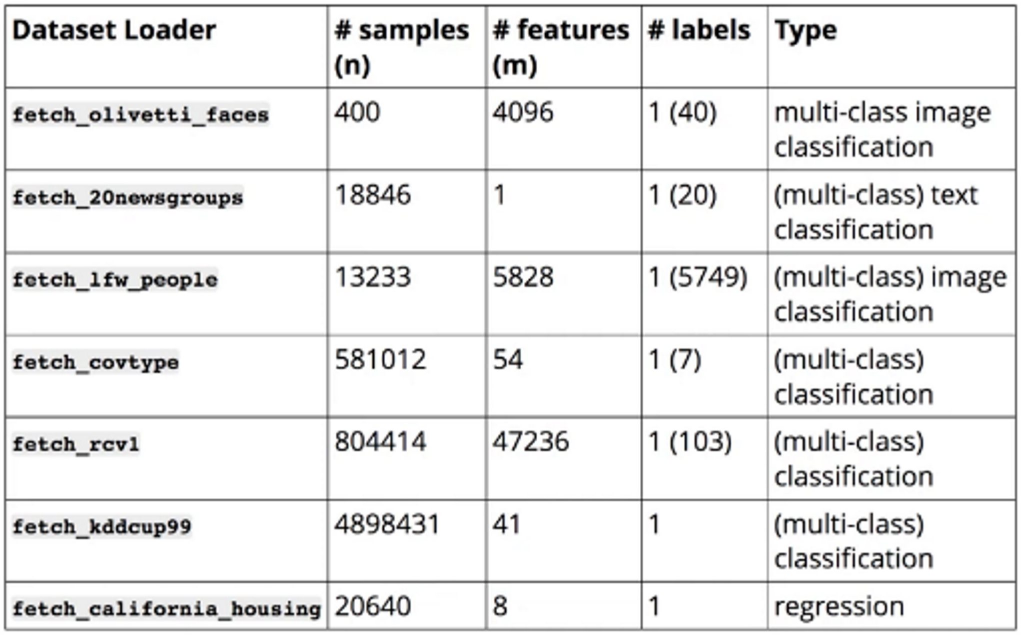
Dataset Loaders



Dataset Loader	# samples (n)	# features (m)	# labels	Type
`load_iris`	150	3	1	Classification
`load_diabetes`	442	10	1	Regression
`load_digits`	1797	64	1	Classification
`load_linnerud`	20	3	3	Regression (multi output)
`load_wine`	178	13	1	Classification
`load_breast_cancer`	569	30	1	Classification

Note: These datasets are bundled with sklearn and we do not require to download them from external sources.

Dataset Fetchers



Dataset Loader	# samples (n)	# features (m)	# labels	Type
`fetch_olivetti_faces`	400	4096	1 (40)	multi-class image classification
`fetch_20newsgroups`	18846	1	1 (20)	(multi-class) text classification
`fetch_lfw_people`	13233	5828	1 (5749)	(multi-class) image classification
`fetch_covtype`	581012	54	1 (7)	(multi-class) classification
`fetch_rcv1`	804414	47236	1 (103)	(multi-class) classification
`fetch_kddcup99`	4898431	41	1	(multi-class) classification
`fetch_california_housing`	20640	8	1	regression

Dataset generators

Regression

`make_regression()` produces regression targets as a sparse random linear combination of random features with noise. The informative features are either uncorrelated or low rank.

Classification

Single label

`make_blobs()` and `make_classification()` first creates a bunch of normally-distributed clusters of points and then assign one or more clusters to each class thereby creating multi-class datasets.

Multilabel

`make_multilabel_classification()` generates random samples with multiple labels with a specific generative process and rejection sampling.

Dataset generators

Clustering

`make_blobs()`generates a bunch of normally-distributed clusters of points with specific mean and standard deviations for each cluster.

Loading external datasets

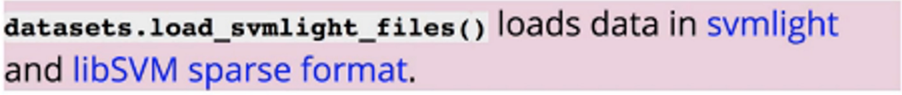
`fetch_openml()`fetches datasets from openml.org, which is a public repository for machine learning data and experiments.

`pandas.io` provides tools to read from common formats like CSV, excel, json, SQL.

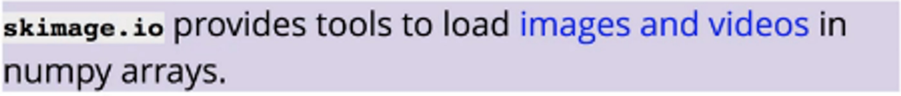
`scipy.io` specializes in binary formats used in scientific computing like .mat and .arff.

`numpy/routines.io` specializes in loading columnar data into numpy arrays.

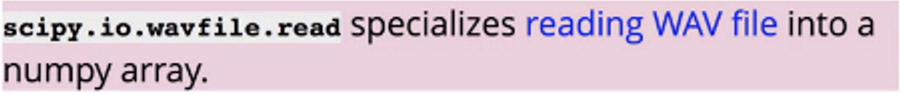
`dataset.load_files` loads directories of text files where directory name is a label and each file is a sample.



`datasets.load_svmlight_files()` loads data in svmlight and libSVM sparse format.



`skimage.io` provides tools to load images and videos in numpy arrays.



`scipy.io.wavfile.read` specializes reading WAV file into a numpy array.

For managing numerical data, sklearn recommends using an optimized file format such as HDF5 (Hierarchical Data Format version 5) to reduce data load times.

Pandas, Py Tables and H5Py provides an interface to read and write data in that format.