Standard Datasets and Basic Functions in Sklearn

ML03

by Prof.Weijia Jia (賈維嘉) N21-G005

Email: jiawj@umac.mo

Standard Datasets in Sklearn
数据文件地址：

（1）https://pan.baidu.com/s/1ZkP4kNRZLEcO71mUJ4zbCw

（2）
https://github.com/liutian111111/Machine-Learning-Course.git

Dataset Overview

	Name	Call Method	Applicable Algorithm	Data Size
Small Dataset	Boston House Price Dataset	load_boston()	regress	506*13
	Iris Flower Dataset	load_iris()	classification	150*4
	Diabetes Dataset	load_diabetes()	regress	442*10
	Handwritten Digital Dataset	load_digits()	classification	5620*64
Big Dataset	Olivetti Facial Image Dataset	fetch_olivetti_faces()	dimensionality reduction	400*64*64
	News Classification Dataset	fetch_20newsgroups()	classification	-
	Labeled Face Dataset	fetch_lfw_people()	classification; dimensionality reduction	-
	Reuters News Corpus Dataset	fetch_recv1()	classification	804414*47236

Note: Small datasets can be used directly, and large datasets should be downloaded automatically at the time of the call (once).

Boston House Price Dataset

- Boston House Price Dataset contains 506 sets of data, each containing details of the house and the surrounding area：
  1. urban crime rates,
  2. nitric oxide concentrations,
  3. average residential homes,
  4. weighted distances to central areas,
  5. average home prices.
  6. 。 。 。

- Boston House price data set can be applied to regression issues.

Boston House Price Dataset

Input

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4

Output

Partial Price Data

Boston House Price Data Set - Property

- CRIM: Urban per capita crime rate
- ZN: Proportion of residential land over 25,000 sq.ft.
- INDUS: Proportion of urban non-retailer land
- CHAS: Charles River empty variable (1 if the boundary is a river; otherwise 0)
- NOX: Nitric oxide concentration
- RM: The average number of rooms in the house.
- AGE: Proportion of self-use houses built before 1940.
- DIS: Weighted distance to five central areas of Boston.
- RAD: The proximity index of a radiating road.
- TAX: A full-value property tax rate of $10,000.
- PTRATIO: The proportion of teachers and students in the town.
- B: 1000(Bk-0.63)^ 2, where Bk refers to the proportion of blacks in the town.
- LSTAT: The proportion of people with low status in the population.
- MEDV: The average house price for a home is in thousands of dollars.

Boston House Price Dataset

Load related datasets using sklearn.datasets.load_boston

Its important parameters are:

- return_X_y: indicates whether to return the target (that is, the price), the default is False, only return data (that is, the attribute).

Boston House Price Data Set - Loading Example

- Example1:

```
>>> from sklearn.datasets import load_boston
>>> boston = load_boston()
>>> print(boston.data.shape)
(506, 13)
```

- Example 2:

```
>>> from sklearn.datasets import load_boston
>>> data, target = load_boston(return_X_y=True)
>>> print(data.shape)
(506, 13)
>>> print(target.shape)
(506)
```

Iris Flower Dataset

- Iris flower dataset collects the measurement data of the iris and the category to which it belongs.
- Measurement data includes: sepal length, sepal width, petal length, and petal width.
- The categories fall into three : Iris Setosa, Iris Versicolour, Iris Virginica. This data set can be used for multi-classification problems.

萼片长度	萼片宽度	花瓣长度	花瓣宽度	类别
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa
4.8	3	1.4	0.1	Iris-setosa
4.3	3	1.1	0.1	Iris-setosa
5.8	4	1.2	0.2	Iris-setosa

Example of the data collection of the Iris flower part

Iris Flower Dataset

Load related datasets using sklearn.datasets. load_iris

Its parameters are:

- return_X_y: If True, the data is returned as (data, target); the default is False, which means that all information (including data and target) is returned in dictionary form.

Iris Flower Dataset - Loading Example

- Example

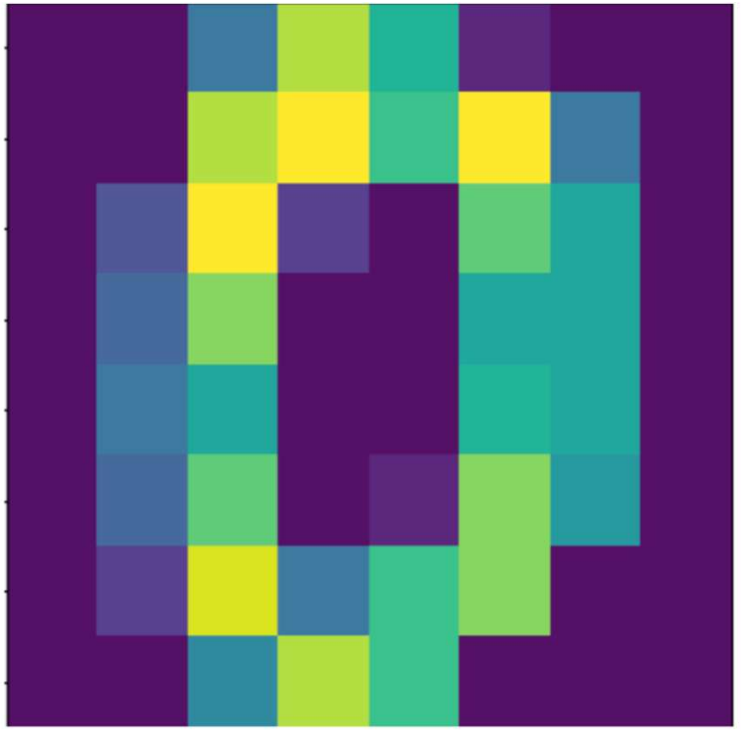
```
>>> from sklearn.datasets import load_iris
>>> iris = load_iris()
>>> print(iris.data.shape)
(150, 4)
>>> print(iris.target.shape)
(150, )
>>> list(iris.target_names)
['setosa', 'versicolor', 'virginica']
```

Handwritten Digital Dataset

- The handwritten digital data set consists of 1797 handwritten digit data of 0-9, each number consisting of a matrix of 8*8 size, the value of the matrix is 0-16, representing the depth of the color.

Handwritten Digital Dataset

0	0	5	13	9	1	0	0
0	0	13	15	10	15	5	0
0	3	15	2	0	11	8	0
0	4	12	0	0	8	8	0
0	5	8	0	0	9	8	0
0	4	11	0	1	12	7	0
0	2	14	5	10	12	0	0
0	0	6	13	10	0	0	0



Sample of number 0

Handwritten Digital Dataset

Load related datasets using sklearn.datasets.load_digits

Its parameters include:

- return_X_y: If True, return data as (data, target); default is False, which means that all information (including data and target) is returned in dictionary form.
- N_class: indicates the number of categories of data returned, such as: n_class=5, returns 0 to 4 data samples.

Handwritten Digital Dataset

Example:

```
>>> from sklearn.datasets import load_digits
>>> digits = load_digits()
>>> print(digits.data.shape)
(1797, 64)
>>> print(digits.target.shape)
 (1797, )
>>> print(digits.images.shape)
 (1797, 8, 8)
>>> import matplotlib.pyplot as plt
 >>> plt.matshow(digits.images[0])
 >>> plt.show()
```

Basic Functions of Sklearn

Basic Functions of Sklearn

- The sklearn library is divided into six parts, which are used to complete classification tasks, regression tasks, clustering tasks, dimensionality reduction tasks, model selection, and data preprocessing.

Classification Task

Classification Task	Loading Module
Nearest Neighbor Algorithm	neighbors.NearestNeighbors
Support Vector Machines	svm.SVC
Naive Bayes	naive_bayes.GaussianNB
Decision Tree	tree.DecisionTreeClassifier
Integration Method	ensemble.BaggingClassifier
Neural Networks	neural_network.MLPClassifier

Regression Task

Regression Task	Loading Module
Ridge Regression	linear_model.Ridge
Lasso Regression	linear_model.Lasso
Flexible Network	linear_model.ElasticNet
Minimum Angle Regression	linear_model.Lars
Bayesian Regression	linear_model.BayesianRidge
Logistic Regression	linear_model.LogisticRegression
Polynomial Regression	preprocessing. PolynomialFeatures

Clustering Task

Clustering Task	Loading Module
K-means	cluster.KMeans
AP Clustering	cluster.AffinityPropagation
Mean Shift	cluster.MeanShift
Hierarchical Clustering	cluster.AgglomerativeClustering
DBSCAN	cluster.DBSCAN
BIRCH	cluster.Birch
Spectral Clustering	cluster.SpectralClustering

Dimensionality Reduction Task

Clustering Task	Loading Module
Principal Component Analysis	decomposition.PCA
Truncating SVD and LSA	decomposition.TruncatedSVD
Dictionary Learning	decomposition.SparseCoder
Factor Analysis	decomposition.FactorAnalysis
Independent Component Analysis	decomposition.FastICA
Non-negative Matrix Factorization	decomposition.NMF
LDA	decomposition.LatentDirichletAllocation