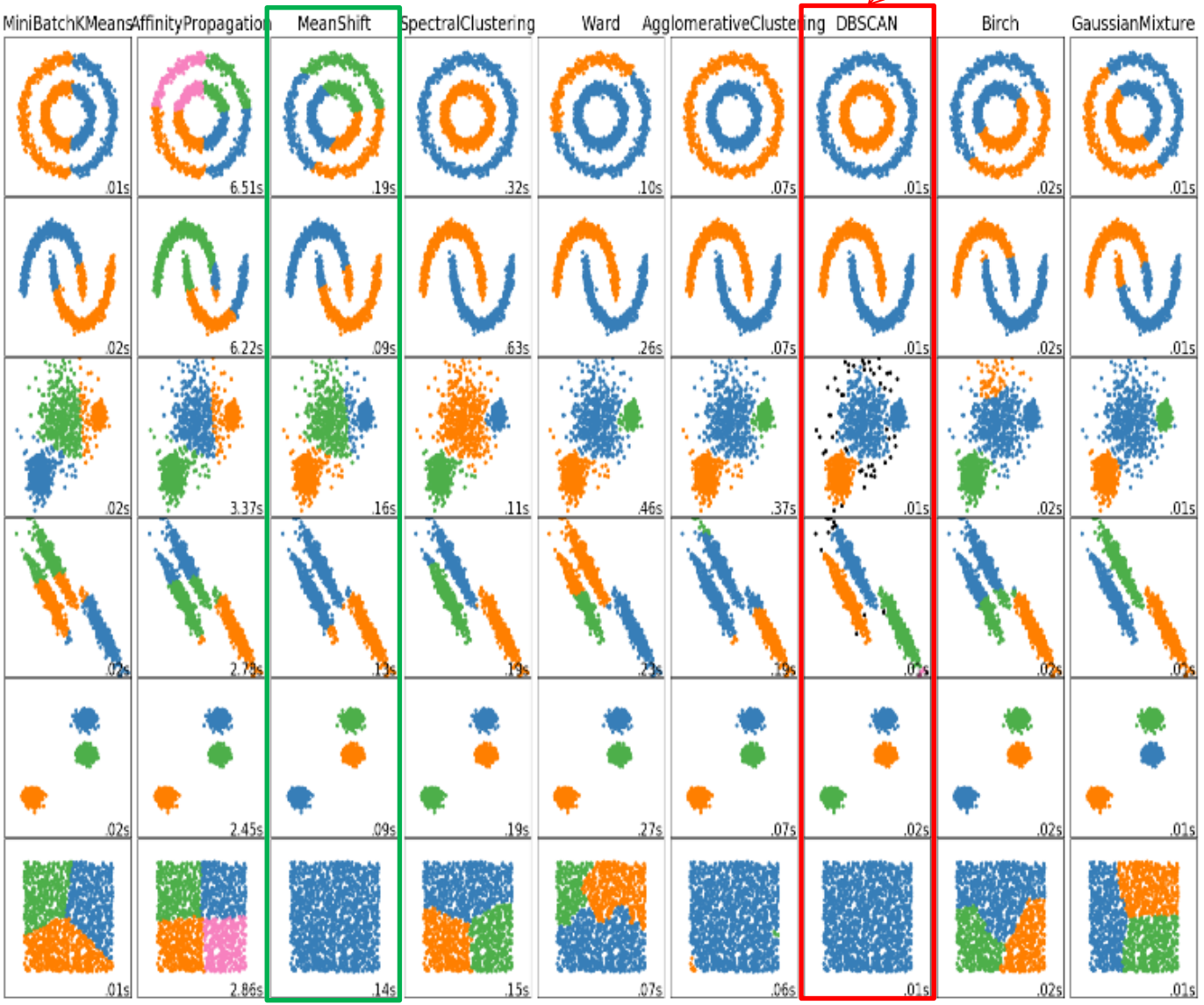
各種クラスタリングの実行結果

DBSCANはクラスタリング良好

各種
パターン

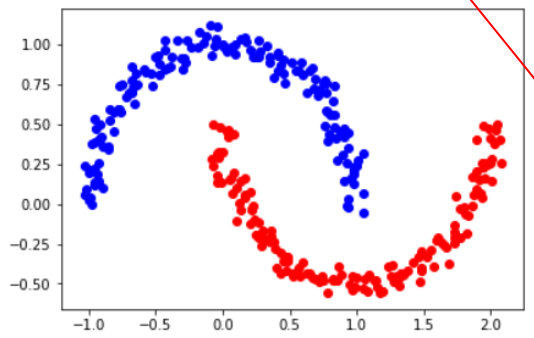


A comparison of the clustering algorithms in scikit-learn

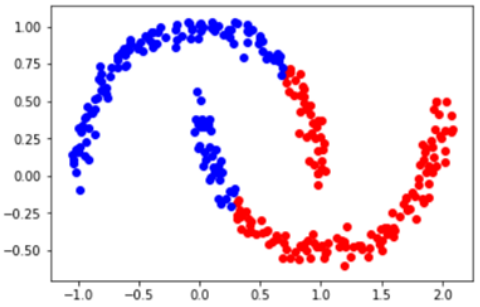
出典：https://scikit-learn.org/stable/modules/clustering.html

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

```
from sklearn.cluster import DBSCAN
X,y = datasets.make_moons(n_samples=300,noise=.05)
db = DBSCAN(eps=0.3, )
db.fit(X)
plt.scatter(X[db.labels_==0,0],X[db.labels_==0,1],c="red")
plt.scatter(X[db.labels_==1,0],X[db.labels_==1,1],c="blue")
plt.show()
```



```
X,y = datasets.make_moons(n_samples=300,noise=.05)
kmeans = KMeans(n_clusters=2)
kmeans.fit(X)
plt.scatter(X[kmeans.labels_==0,0],X[kmeans.labels_==0,1],c="red")
plt.scatter(X[kmeans.labels_==1,0],X[kmeans.labels_==1,1],c="blue")
plt.show()
```



sklearn.cluster : Clustering

The `sklearn.cluster` module gathers popular unsupervised clustering algorithms.

User guide: See the Clustering section for further details.

Classes

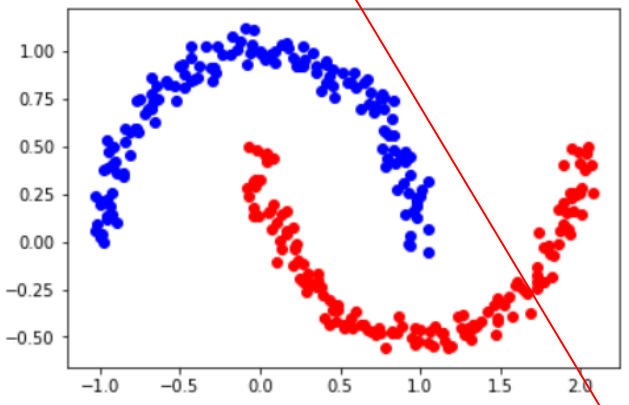
`cluster.AffinityPropagation` ([damping, ...])	Perform Affinity Propagation Clustering of data.
`cluster.AgglomerativeClustering` ([...])	Agglomerative Clustering
`cluster.Birch` ([threshold, branching_factor, ...])	Implements the Birch clustering algorithm.
`cluster.DBSCAN` ([eps, min_samples, metric, ...])	Perform DBSCAN clustering from vector array or distance matrix.
`cluster.FeatureAgglomeration` ([n_clusters, ...])	Agglomerate features.
`cluster.KMeans` ([n_clusters, init, n_init, ...])	K-Means clustering
`cluster.MiniBatchKMeans` ([n_clusters, init, ...])	Mini-Batch K-Means clustering
`cluster.MeanShift` ([bandwidth, seeds, ...])	Mean shift clustering using a flat kernel.
`cluster.SpectralClustering` ([n_clusters, ...])	Apply clustering to a projection to the normalized laplacian.

Functions

`cluster.affinity_propagation` (S[, ...])	Perform Affinity Propagation Clustering of data
`cluster.dbscan` (X [, eps, min_samples, ...])	Perform DBSCAN clustering from vector array or distance matrix.
`cluster.estimate_bandwidth` (X [, quantile, ...])	Estimate the bandwidth to use with the mean-shift algorithm.
`cluster.k_means` (X, n_clusters[, ...])	K-means clustering algorithm.
`cluster.mean_shift` (X [, bandwidth, seeds, ...])	Perform mean shift clustering of data using a flat kernel.
`cluster.spectral_clustering` (affinity[, ...])	Apply clustering to a projection to the normalized laplacian.
`cluster.ward_tree` (X [, connectivity, ...])	Ward clustering based on a Feature matrix.

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

```
from sklearn.cluster import DBSCAN
X,y = datasets.make_moons(n_samples=300,noise=.05)
db = DBSCAN(eps=0.3, )
db.fit(X)
plt.scatter(X[db.labels_==0,0],X[db.labels_==0,1],c="red")
plt.scatter(X[db.labels_==1,0],X[db.labels_==1,1],c="blue")
plt.show()
```



dbラベル0の0列目をx、
1列目をyにして
赤色で散布図

dbラベル1の0列目をx、
1列目をyにして
青色で散布図

他のｱﾙｺﾞﾘｽﾞﾑの場合、
dbのところを変える

sklearn.datasets : Datasets

The sklearn.datasets module includes utilities to load datasets, including methods to load and fetch popular reference datasets. It also features some artificial data generators.

User guide: See the Dataset loading utilities section for further details.

Loaders

datasets.clear_data_home([data_home])	Delete all the content of the data home cache.
datasets.dump_svmlight_file(X, y, f[, ...])	Dump the dataset in svmlight / libsvm file format.
datasets.fetch_20newsgroups([data_home, ...])	Load the filenames and data from the 20 newsgroups dataset (classification).
datasets.fetch_20newsgroups_vectorized([...])	Load the 20 newsgroups dataset and vectorize it into token counts (classification).
datasets.fetch_california_housing([...])	Load the California housing dataset (regression).
datasets.fetch_covtype([data_home, ...])	Load the covertype dataset (classification).

Samples generator

datasets.make_biclusters(shape, n_clusters)	Generate an array with constant block diagonal structure for biclustering.
datasets.make_blobs([n_samples, n_features, ...])	Generate isotropic Gaussian blobs for clustering.
datasets.make_checkerboard(shape, n_clusters)	Generate an array with block checkerboard structure for biclustering.
datasets.make_circles([n_samples, shuffle, ...])	Make a large circle containing a smaller circle in 2d.
datasets.make_classification([n_samples, ...])	Generate a random n-class classification problem.
datasets.make_friedman1([n_samples, ...])	Generate the “Friedman #1” regression problem
datasets.make_friedman2([n_samples, noise, ...])	Generate the “Friedman #2” regression problem
datasets.make_friedman3([n_samples, noise, ...])	Generate the “Friedman #3” regression problem
datasets.make_gaussian_quantiles([mean, ...])	Generate isotropic Gaussian and label samples by quantile
datasets.make_hastie_10_2([n_samples, ...])	Generates data for binary classification used in Hastie et al.
datasets.make_low_rank_matrix([n_samples, ...])	Generate a mostly low rank matrix with bell-shaped singular values
datasets.make_moons([n_samples, shuffle, ...])	Make two interleaving half circles
datasets.make_multilabel_classification([...])	Generate a random multilabel classification problem.
datasets.make_regression([n_samples, ...])	Generate a random regression problem.
datasets.make_s_curve([n_samples, noise, ...])	Generate an S curve dataset.
datasets.make_sparse_coded_signal(n_samples, ...)	Generate a signal as a sparse combination of dictionary elements.
datasets.make_sparse_spd_matrix([dim, ...])	Generate a sparse symmetric definite positive matrix.
datasets.make_sparse_uncorrelated([...])	Generate a random regression problem with sparse uncorrelated design
datasets.make_spd_matrix(n_dim[, random_state])	Generate a random symmetric, positive-definite matrix.
datasets.make_swiss_roll([n_samples, noise, ...])	Generate a swiss roll dataset.