

## Оглавление

sklearn.base: Base classes and utility functions.....	2
Base classes.....	2
Functions.....	3
sklearn.calibration: Probability Calibration.....	3
sklearn.cluster: Clustering.....	3
Classes.....	3
Functions.....	4
sklearn.cluster.bicluster: Biclustering.....	4
Classes.....	4
sklearn.compose: Composite Estimators.....	5
sklearn.covariance: Covariance Estimators.....	5
sklearn.cross_decomposition: Cross decomposition.....	6
sklearn.datasets: Datasets.....	6
Loaders.....	6
Samples generator.....	8
sklearn.decomposition: Matrix Decomposition.....	9
sklearn.discriminant_analysis: Discriminant Analysis.....	10
sklearn.dummy: Dummy estimators Фиктивные оценки.....	10
sklearn.ensemble: Ensemble Methods.....	10
sklearn.exceptions: Exceptions and warnings.....	11
sklearn.experimental: Experimental.....	12
sklearn.feature_extraction: Feature Extraction.....	12
From images.....	13
From text.....	13
sklearn.feature_selection: Feature Selection.....	13
sklearn.gaussian_process: Gaussian Processes.....	15
sklearn.isotonic: Isotonic regression.....	15
sklearn.impute: Impute.....	15
sklearn.kernel_approximation Kernel Approximation.....	15
sklearn.kernel_ridge Kernel Ridge Regression.....	16
sklearn.linear_model: Generalized Linear Models.....	16
sklearn.manifold: Manifold Learning.....	18
Многообразие обучения.....	18
sklearn.metrics: Metrics.....	19
Model Selection Interface.....	19
Classification metrics.....	19
Regression metrics.....	20
Multilabel ranking metrics.....	21
Clustering metrics.....	21
Biclustering metrics.....	22
Pairwise metrics.....	22
sklearn.mixture: Gaussian Mixture Models.....	23
sklearn.model_selection: Model Selection.....	23
Splitter Classes.....	24
Splitter Functions.....	24
Hyper-parameter optimizers.....	24
Model validation.....	25
sklearn.multiclass: Multiclass and multilabel classification.....	25

Multiclass and multilabel classification strategies.....	25
sklearn.multioutput: Multioutput regression and classification.....	26
sklearn.naive_bayes: Naive Bayes.....	27
sklearn.neighbors: Nearest Neighbors.....	27
sklearn.neural_network: Neural network models.....	28
sklearn.pipeline: Pipeline.....	28
sklearn.inspection: inspection.....	29
sklearn.preprocessing: Preprocessing and Normalization.....	29
sklearn.random_projection: Random projection.....	30
sklearn.semi_supervised Semi-Supervised Learning.....	31
sklearn.svm: Support Vector Machines.....	32
Estimators.....	32
Low-level methods.....	32
sklearn.tree: Decision Trees.....	32
sklearn.utils: Utilities.....	33
Recently deprecated Недавно устарел.....	36
To be removed in 0.23 Быть удаленным в 0.23.....	36
To be removed in 0.22 Быть удаленным в 0.22.....	37

This is the class and function reference of scikit-learn. Please refer to the [full user guide](#) for further details, as the class and function raw specifications may not be enough to give full guidelines on their uses. For reference on concepts repeated across the API, see [Glossary of Common Terms and API Elements](#).

Это ссылка на класс и функцию scikit-learn. Пожалуйста, обратитесь к полному руководству пользователя для получения более подробной информации, поскольку необработанных спецификаций класса и функции может быть недостаточно, чтобы дать полное руководство по их использованию. Ссылка на понятия, повторяемые в API, приведена в Глоссарии общих терминов и элементов API.

## **sklearn.base: Base classes and utility functions**

Base classes for all estimators. Базовые классы для всех оценщиков

### **Base classes**

<a href="#">base.BaseEstimator</a>	Base class for all estimators in scikit-learn
<a href="#">base.BiclusterMixin</a>	Mixin class for all bicluster estimators in scikit-learn Класс Mixin для всех бикластерных оценок в scikit-learn
<a href="#">base.ClassifierMixin</a>	Mixin class for all classifiers in scikit-learn.
<a href="#">base.ClusterMixin</a>	Mixin class for all cluster estimators in scikit-learn.
<a href="#">base.DensityMixin</a>	Mixin class for all density estimators in scikit-learn. Класс Mixin для всех оценщиков плотности в scikit-learn.
<a href="#">base.RegressorMixin</a>	Mixin class for all regression estimators in scikit-learn.
<a href="#">base.TransformerMixin</a>	Mixin class for all transformers in scikit-learn.

## Functions

<a href="#">base.clone</a> (estimator[, safe])	Constructs a new estimator with the same parameters. Создает новый оценщик с теми же параметрами.
<a href="#">base.is_classifier</a> (estimator)	Returns True if the given estimator is (probably) a classifier. Возвращает True, если данный оценщик (вероятно) является классификатором.
<a href="#">base.is_regressor</a> (estimator)	Returns True if the given estimator is (probably) a regressor.
<a href="#">config_context</a> (\(*\*new_config)	Context manager for global scikit-learn configuration
<a href="#">get_config</a> ()	Retrieve current values for configuration set by <a href="#">set_config</a> Получить текущие значения для конфигурации, установленной <a href="#">set_config</a>
<a href="#">set_config</a> ([assume_finite, working_memory, ...])	Set global scikit-learn configuration
<a href="#">show_versions</a> ()	Print useful debugging information Распечатать полезную информацию об отладке

## [\*\*sklearn.calibration: Probability Calibration\*\*](#)

Calibration of predicted probabilities. Калибровка прогнозируемых вероятностей.

**User guide:** See the [Probability calibration](#) section for further details.

<a href="#">calibration.CalibratedClassifierCV</a> (...)	Probability calibration with isotonic regression or sigmoid. Калибровка вероятности с изотонической регрессией или сигмOIDом.
<a href="#">calibration.calibration_curve</a> (y_true, y_prob)	Compute true and predicted probabilities for a calibration curve. Вычислить истинные и прогнозируемые вероятности для калибровочной кривой.

## [\*\*sklearn.cluster: Clustering\*\*](#)

The [sklearn.cluster](#) module gathers popular unsupervised clustering algorithms.

Модуль sklearn.cluster собирает популярные неконтролируемые алгоритмы кластеризации.

**User guide:** See the [Clustering](#) section for further details.

## Classes

<a href="#">cluster.AffinityPropagation</a> ([damping, ...])	Perform Affinity Propagation Clustering of data. Выполнить кластеризацию данных по Affinity Propagation.
<a href="#">cluster.AgglomerativeClustering</a> (...)	Agglomerative Clustering
<a href="#">cluster.Birch</a> ([threshold, branching_factor, ...])	Implements the Birch clustering algorithm. Реализует алгоритм кластеризации березы.
<a href="#">cluster.DBSCAN</a> ([eps, min_samples, metric, ...])	Perform DBSCAN clustering from vector array or distance matrix. Выполните кластеризацию DBSCAN из векторного массива или матрицы расстояний.
<a href="#">cluster.OPTICS</a> ([min_samples, max_eps, ...])	Estimate clustering structure from vector array Оценить структуру кластеризации из векторного массива

<a href="#"><code>cluster.FeatureAgglomeration</code></a> ([n_clusters, ...])	Agglomerate features. Особенности агломерата.
<a href="#"><code>cluster.KMeans</code></a> ([n_clusters, init, n_init, ...])	K-Means clustering
<a href="#"><code>cluster.MinibatchKMeans</code></a> ([n_clusters, init, ...])	Mini-Batch K-Means clustering
<a href="#"><code>cluster.MeanShift</code></a> ([bandwidth, seeds, ...])	Mean shift clustering using a flat kernel. Кластеризация среднего сдвига с использованием плоского ядра.
<a href="#"><code>cluster.SpectralClustering</code></a> ([n_clusters, ...])	Apply clustering to a projection of the normalized Laplacian. Примените кластеризацию к проекции нормированного лапласиана.

## Functions

<a href="#"><code>cluster.affinity_propagation</code></a> (S[, ...])	Perform Affinity Propagation Clustering of data
<a href="#"><code>cluster.cluster_optics_dbscan</code></a> (reachability, ...)	Performs DBSCAN extraction for an arbitrary epsilon. Выполняет извлечение DBSCAN для произвольного эпсилона.
<a href="#"><code>cluster.cluster_optics_xi</code></a> (reachability, ...)	Automatically extract clusters according to the Xi-steep method. Автоматически извлекать кластеры по Xi-крутому методу.
<a href="#"><code>cluster.compute_optics_graph</code></a> (X, min_samples, ...)	Computes the OPTICS reachability graph. Вычисляет график достижимости OPTICS.
<a href="#"><code>cluster.dbscan</code></a> (X[, eps, min_samples, ...])	Perform DBSCAN clustering from vector array or distance matrix.
<a href="#"><code>cluster.estimate_bandwidth</code></a> (X[, quantile, ...])	Estimate the bandwidth to use with the mean-shift algorithm. Оцените пропускную способность для использования с алгоритмом среднего сдвига.
<a href="#"><code>cluster.k_means</code></a> (X, n_clusters[, ...])	K-means clustering algorithm.
<a href="#"><code>cluster.mean_shift</code></a> (X[, bandwidth, seeds, ...])	Perform mean shift clustering of data using a flat kernel. Выполните среднюю смену кластеризации данных, используя плоское ядро.
<a href="#"><code>cluster.spectral_clustering</code></a> (affinity[, ...])	Apply clustering to a projection of the normalized Laplacian. Примените кластеризацию к проекции нормированного лапласиана.
<a href="#"><code>cluster.ward_tree</code></a> (X[, connectivity, ...])	Ward clustering based on a Feature matrix. Кластерная палата на основе матрицы объектов.

## **sklearn.cluster.bicluster: Biclustering**

Spectral biclustering algorithms. Спектральные алгоритмы бикластеризации.

Authors : Kemal Eren License: BSD 3 clause

**User guide:** See the [Biclustering](#) section for further details.

## Classes

<a href="#"><code>SpectralBiclustering</code></a> ([n_clusters, method, ...])	Spectral biclustering (Kluger, 2003).
<a href="#"><code>SpectralCoclustering</code></a> ([n_clusters, ...])	Spectral Co-Clustering algorithm (Dhillon, 2001).

## sklearn.compose: Composite Estimators

Meta-estimators for building composite models with transformers

In addition to its current contents, this module will eventually be home to refurbished versions of Pipeline and FeatureUnion.

Мета-оценщики для построения составных моделей с трансформаторами

В дополнение к его текущему содержанию, этот модуль в конечном итоге станет домом для обновленных версий Pipeline и FeatureUnion.

**User guide:** See the [Pipelines and composite estimators](#) section for further details.

<a href="#">compose.ColumnTransformer(transformers[, ...])</a>	Applies transformers to columns of an array or pandas DataFrame.
<a href="#">compose.TransformedTargetRegressor([...])</a>	Meta-estimator to regress on a transformed target.
<a href="#">compose.make_column_transformer(...)</a>	Construct a ColumnTransformer from the given transformers. Построить ColumnTransformer из заданных трансформаторов.

## sklearn.covariance: Covariance Estimators

The [sklearn.covariance](#) module includes methods and algorithms to robustly estimate the covariance of features given a set of points. The precision matrix defined as the inverse of the covariance is also estimated. Covariance estimation is closely related to the theory of Gaussian Graphical Models.

Модуль sklearn.covariance включает методы и алгоритмы, позволяющие надежно оценить ковариантность объектов с учетом набора точек. Матрица точности, определенная как обратная ковариация, также оценивается. Оценка ковариации тесно связана с теорией гауссовых графических моделей.

**User guide:** See the [Covariance estimation](#) section for further details.

<a href="#">covariance.EmpiricalCovariance([...])</a>	Maximum likelihood covariance estimator Оценщик максимальной вероятности ковариации
<a href="#">covariance.EllipticEnvelope([...])</a>	An object for detecting outliers in a Gaussian distributed dataset. Объект для обнаружения выбросов в гауссовском распределенном наборе данных.
<a href="#">covariance.GraphicalLasso([alpha, mode, ...])</a>	Sparse inverse covariance estimation with an l1-penalized estimator. Оценка разреженной обратной ковариации с помощью l1-штрафной оценки.
<a href="#">covariance.GraphicalLassoCV([alphas, ...])</a>	Sparse inverse covariance w/ cross-validated choice of the l1 penalty. Разреженная обратная ковариация с перекрестным подтверждением выбора штрафа l1.
<a href="#">covariance.LedoitWolf([store_precision, ...])</a>	LedoitWolf Estimator
<a href="#">covariance.MinCovDet([store_precision, ...])</a>	Minimum Covariance Determinant (MCD): robust estimator of covariance. Определитель минимальной ковариации (MCD): надежная оценка ковариации.
<a href="#">covariance.OAS([store_precision, ...])</a>	Oracle Approximating Shrinkage Estimator Oracle приближается к оценке усадки
<a href="#">covariance.ShrunkCovariance([...])</a>	Covariance estimator with shrinkage Ковариационная оценка с усадкой

<a href="#"><code>covariance.empirical_covariance(X[, ...])</code></a>	Computes the Maximum likelihood covariance estimator Вычисляет ковариационную оценку максимального правдоподобия
<a href="#"><code>covariance.graphical_lasso(emp_cov, alpha[, ...])</code></a>	l1-penalized covariance estimator l1-оштрафованная оценка ковариации
<a href="#"><code>covariance.ledoit_wolf(X[, assume_centered, ...])</code></a>	Estimates the shrunk Ledoit-Wolf covariance matrix.
<a href="#"><code>covariance.oas(X[, assume_centered])</code></a>	Estimate covariance with the Oracle Approximating Shrinkage algorithm. Оцените ковариацию с помощью алгоритма аппроксимирующей усадки Oracle.
<a href="#"><code>covariance.shrunk_covariance(emp_cov[, ...])</code></a>	Calculates a covariance matrix shrunk on the diagonal Вычисляет ковариационную матрицу, сжатую по диагонали

## **sklearn.cross\_decomposition: Cross decomposition**

### **Перекрестное разложение**

**User guide:** See the [Cross decomposition](#) section for further details.

<a href="#"><code>cross_decomposition.CCA([n_components, ...])</code></a>	CCA Canonical Correlation Analysis. CCA Канонический корреляционный анализ.
<a href="#"><code>cross_decomposition.PLSCanonical([...])</code></a>	PLS Canonical implements the 2 blocks canonical PLS of the original Wold algorithm [Tenenhaus 1998] p.204, referred as PLS-C2A in [Wegelin 2000]. PLS Canonical реализует 2 блока канонического PLS оригинального алгоритма Wold
<a href="#"><code>cross_decomposition.PLSRegression([...])</code></a>	PLS regression PLS Регрессия
<a href="#"><code>cross_decomposition.PLSVD([n_comp</code></a>	Partial Least Square SVD

## **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.

Модуль `sklearn.datasets` содержит утилиты для загрузки наборов данных, в том числе методы для загрузки и выборки популярных эталонных наборов данных. Это также показывает некоторые искусственные генераторы данных.

**User guide:** See the [Dataset loading utilities](#) section for further details.

### **Loaders**

<a href="#"><code>datasets.clear_data_home([data_home])</code></a>	Delete all the content of the data home cache. Удалите все содержимое данных домашнего кэша.
<a href="#"><code>datasets.dump_svmlight_file(X, y, f[, ...])</code></a>	Dump the dataset in svmlight / libsvm file format. Дамп набора данных в формате файла svmlight / libsvm.
<a href="#"><code>datasets.fetch_20newsgroups([data_home, ...])</code></a>	Load the filenames and data from the 20 newsgroups dataset (classification). Загрузите имена файлов и данные из набора данных 20 групп новостей (классификация).
<a href="#"><code>datasets.fetch_20newsgroups_vectorized([...])</code></a>	Load the 20 newsgroups dataset and vectorize it into token counts (classification). Загрузите набор данных из 20 групп новостей и векторизуйте его в число токенов (классификация).

<a href="#"><code>datasets.fetch_california_housing(...)</code></a>	Load the California housing dataset (regression). Загрузите набор данных о жилье в Калифорнии (регрессия).
<a href="#"><code>datasets.fetch_covtype([data_home, ...])</code></a>	Load the covtype dataset (classification). Загрузите набор данных обложки (классификация).
<a href="#"><code>datasets.fetch_kddcup99([subset, data_home, ...])</code></a>	Load the kddcup99 dataset (classification). Загрузите набор данных kddcup99 (классификация).
<a href="#"><code>datasets.fetch_lfw_pairs([subset, ...])</code></a>	Load the Labeled Faces in the Wild (LFW) pairs dataset (classification). Загрузите набор пар «Помеченные грани в дикой природе» (классификация).
<a href="#"><code>datasets.fetch_lfw_people([data_home, ...])</code></a>	Load the Labeled Faces in the Wild (LFW) people dataset (classification). Загрузите набор данных «Помеченные лица людей в дикой природе» (классификация).
<a href="#"><code>datasets.fetch_olivetti_faces([data_home, ...])</code></a>	Load the Olivetti faces data-set from AT&T (classification). Загрузите набор данных Olivetti Faces из AT & T (классификация).
<a href="#"><code>datasets.fetch_openml([name, version, ...])</code></a>	Fetch dataset from openml by name or dataset id. Получить набор данных из openml по имени или идентификатору набора данных.
<a href="#"><code>datasets.fetch_rcv1([data_home, subset, ...])</code></a>	Load the RCV1 multilabel dataset (classification). Загрузите многослойный набор данных RCV1 (классификация).
<a href="#"><code>datasets.fetch_species_distributions(...)</code></a>	Loader for species distribution dataset from Phillips et. Загрузчик для набора данных о распределении видов от Phillips et.
<a href="#"><code>datasets.get_data_home([data_home])</code></a>	Return the path of the scikit-learn data dir. Возвращает путь к каталогу данных scikit-learn.
<a href="#"><code>datasets.load_boston([return_X_y])</code></a>	Load and return the boston house-prices dataset (regression). Загрузите и верните набор данных по ценам на жилье в Бостоне (регрессия).
<a href="#"><code>datasets.load_breast_cancer([return_X_y])</code></a>	Load and return the breast cancer wisconsin dataset (classification). Загрузите и верните набор данных Висконсин по раку молочной железы (классификация).
<a href="#"><code>datasets.load_diabetes([return_X_y])</code></a>	Load and return the diabetes dataset (regression). Загрузите и верните набор данных о диабете (регрессия).
<a href="#"><code>datasets.load_digits([n_class, return_X_y])</code></a>	Load and return the digits dataset (classification). Загрузите и верните набор данных цифр (классификация).
<a href="#"><code>datasets.load_files(container_path[, ...])</code></a>	Load text files with categories as subfolder names. Загружайте текстовые файлы с категориями в качестве имен подпапок.
<a href="#"><code>datasets.load_iris([return_X_y])</code></a>	Load and return the iris dataset (classification). Загрузите и верните набор данных цветков ириса (классификация).
<a href="#"><code>datasets.load_linnerud([return_X_y])</code></a>	Load and return the linnerud dataset (multivariate regression). Загрузите и верните набор данных linnerud (многомерная регрессия).
<a href="#"><code>datasets.load_sample_image(image_name)</code></a>	Load the numpy array of a single sample image Загрузите массив пустых изображений одного образца



<a href="#"><code>datasets.load_sample_images()</code></a>	Load sample images for image manipulation. Загрузите образцы изображений для работы с изображениями.
<a href="#"><code>datasets.load_svmlight_file(f[, n_features, ...])</code></a>	Load datasets in the svmlight / libsvm format into sparse CSR matrix Загрузка наборов данных в формате svmlight / libsvm в разреженную матрицу CSR
<a href="#"><code>datasets.load_svmlight_files(files[, ...])</code></a>	Load dataset from multiple files in SVMlight format Загрузка набора данных из нескольких файлов в формате SVMlight
<a href="#"><code>datasets.load_wine([return_X_y])</code></a>	Load and return the wine dataset (classification). Загрузите и верните набор данных вина (классификация).

## Samples generator

<a href="#"><code>datasets.make_biclusters(shape, n_clusters)</code></a>	Generate an array with constant block diagonal structure for biclustering. Создайте массив с постоянной диагональной структурой блоков для бикластеризации.
<a href="#"><code>datasets.make_blobs([n_samples, n_features, ...])</code></a>	Generate isotropic Gaussian blobs for clustering. Генерация изотропных гауссовых капель для кластеризации.
<a href="#"><code>datasets.make_checkerboard(shape, n_clusters)</code></a>	Generate an array with block checkerboard structure for biclustering. Создайте массив с блочной структурой шахматной доски для бикластеризации.
<a href="#"><code>datasets.make_circles([n_samples, shuffle, ...])</code></a>	Make a large circle containing a smaller circle in 2d. Сделайте большой круг, содержащий меньший круг в 2d.
<a href="#"><code>datasets.make_classification([n_samples, ...])</code></a>	Generate a random n-class classification problem. Генерация случайной задачи классификации n-класса.
<a href="#"><code>datasets.make_friedman1([n_samples, ...])</code></a>	Generate the “Friedman #1” regression problem Генерация регрессионной задачи «Фридман № 1»
<a href="#"><code>datasets.make_friedman2([n_samples, noise, ...])</code></a>	Generate the “Friedman #2” regression problem
<a href="#"><code>datasets.make_friedman3([n_samples, noise, ...])</code></a>	Generate the “Friedman #3” regression problem
<a href="#"><code>datasets.make_gaussian_quantiles([mean, ...])</code></a>	Generate isotropic Gaussian and label samples by quantile Генерация изотропных гауссовских и метка образцов квантиль
<a href="#"><code>datasets.make_hastie_10_2([n_samples, ...])</code></a>	Generates data for binary classification used in Hastie et al. Генерирует данные для двоичной классификации, используемые в Hastie et al.
<a href="#"><code>datasets.make_low_rank_matrix([n_samples, ...])</code></a>	Generate a mostly low rank matrix with bell-shaped singular values Генерация в основном матрицы низкого ранга с колоколообразными единичными значениями
<a href="#"><code>datasets.make_moons([n_samples, shuffle, ...])</code></a>	Make two interleaving half circles Сделайте два чередующихся полукруга
<a href="#"><code>datasets.make_multilabel_classification([...])</code></a>	Generate a random multilabel classification problem. Создайте случайную задачу классификации с несколькими метками.
<a href="#"><code>datasets.make_regression([n_samples, ...])</code></a>	Generate a random regression problem. Генерация случайной регрессии.
<a href="#"><code>datasets.make_s_curve([n_samples, noise, ...])</code></a>	Generate an S curve dataset. Создайте набор данных кривой S.
<a href="#"><code>datasets.make_sparse_coded_signal(n_samples, ...)</code></a>	Generate a signal as a sparse combination of dictionary elements. Генерируйте сигнал как разреженную



	комбинацию элементов словаря.
<a href="#"><code>datasets.make_sparse_spd_matrix(dim, ...)</code></a>	Generate a sparse symmetric definite positive matrix. Создать разреженную симметричную определенную положительную матрицу.
<a href="#"><code>datasets.make_sparse_uncorrelated(...)</code></a>	Generate a random regression problem with sparse uncorrelated design Генерировать случайную регрессионную проблему с разреженным некоррелированным дизайном
<a href="#"><code>datasets.make_spd_matrix(n_dim[, random_state])</code></a>	Generate a random symmetric, positive-definite matrix. Генерация случайной симметричной положительно определенной матрицы.
<a href="#"><code>datasets.make_swiss_roll(n_samples, noise, ...)</code></a>	Generate a swiss roll dataset. Создайте набор данных швейцарского рулона.

## **sklearn.decomposition: Matrix Decomposition**

The [`sklearn.decomposition`](#) module includes matrix decomposition algorithms, including among others PCA, NMF or ICA. Most of the algorithms of this module can be regarded as dimensionality reduction techniques.

Модуль `sklearn.decomposition` включает в себя алгоритмы матричной декомпозиции, в том числе PCA, NMF или ICA. Большинство алгоритмов этого модуля можно рассматривать как методы уменьшения размерности.

**User guide:** See the [Decomposing signals in components \(matrix factorization problems\)](#) section for further details.

<a href="#"><code>decomposition.DictionaryLearning(...)</code></a>	Dictionary learning Изучение словаря
<a href="#"><code>decomposition.FactorAnalysis(n_components, ...)</code></a>	Factor Analysis (FA) Факторный анализ
<a href="#"><code>decomposition.FastICA(n_components, ...)</code></a>	FastICA: a fast algorithm for Independent Component Analysis. быстрый алгоритм для независимого анализа компонентов
<a href="#"><code>decomposition.IncrementalPCA(n_components, ...)</code></a>	Incremental principal components analysis (IPCA). Инкрементальный анализ основных компонентов
<a href="#"><code>decomposition.KernelPCA(n_components, ...)</code></a>	Kernel Principal component analysis (KPCA) Анализ основных компонентов ядра
<a href="#"><code>decomposition.LatentDirichletAllocation(...)</code></a>	Latent Dirichlet Allocation with online variational Bayes algorithm Латентное распределение Дирихле с помощью онлайн-вариационного алгоритма Байеса
<a href="#"><code>decomposition.MinibatchDictionaryLearning(...)</code></a>	Mini-batch dictionary learning Мини-пакетное изучение словаря
<a href="#"><code>decomposition.MinibatchSparsePCA(...)</code></a>	Mini-batch Sparse Principal Components Analysis Мини-пакетный Разреженный анализ основных компонентов
<a href="#"><code>decomposition.NMF(n_components, init, ...)</code></a>	Non-Negative Matrix Factorization (NMF) Неотрицательная матричная факторизация
<a href="#"><code>decomposition.PCA(n_components, copy, ...)</code></a>	Principal component analysis (PCA) Анализ главных компонентов
<a href="#"><code>decomposition.SparsePCA(n_components, ...)</code></a>	Sparse Principal Components Analysis (SparsePCA) Разреженный анализ главных компонентов

<a href="#"><code>decomposition.SparseCoder</code>(dictionary[, ...])</a>	Sparse coding Разреженное кодирование
<a href="#"><code>decomposition.TruncatedSVD</code>(n_components, ...)</a>	Dimensionality reduction using truncated SVD (aka LSA). Уменьшение размерности с использованием усеченного SVD (он же LSA).
<a href="#"><code>decomposition.dict_learning</code>(X, n_components, ...)</a>	Solves a dictionary learning matrix factorization problem. Решает проблему факторизации матрицы изучения словаря.
<a href="#"><code>decomposition.dict_learning_online</code>(X[, ...])</a>	Solves a dictionary learning matrix factorization problem online. Решает проблему факторизации матрицы изучения словаря онлайн.
<a href="#"><code>decomposition.fastica</code>(X[, n_components, ...])</a>	Perform Fast Independent Component Analysis. Выполните быстрый независимый анализ компонентов.
<a href="#"><code>decomposition.non_negative_factorization</code>(X)</a>	Compute Non-negative Matrix Factorization (NMF) Вычислить неотрицательную матричную факторизацию (NMF)
<a href="#"><code>decomposition.sparse_encode</code>(X, dictionary[, ...])</a>	Sparse coding Разреженное кодирование

## **sklearn.discriminant\_analysis: Discriminant Analysis**

Linear Discriminant Analysis and Quadratic Discriminant Analysis

Линейный дискриминантный анализ и квадратичный дискриминантный анализ

**User guide:** See the [Linear and Quadratic Discriminant Analysis](#) section for further details.

<a href="#"><code>discriminant_analysis.LinearDiscriminantAnalysis</code>(...)</a>	Linear Discriminant Analysis Линейный Дискриминантный Анализ
<a href="#"><code>discriminant_analysis.QuadraticDiscriminantAnalysis</code>(...)</a>	Quadratic Discriminant Analysis Квадратичный Дискриминантный Анализ

## **sklearn.dummy: Dummy estimators Фиктивные оценки**

**User guide:** See the [Model evaluation: quantifying the quality of predictions](#) section for further details.

<a href="#"><code>dummy.DummyClassifier</code>(strategy, ...)</a>	DummyClassifier is a classifier that makes predictions using simple rules. DummyClassifier - это классификатор, который делает прогнозы, используя простые правила.
<a href="#"><code>dummy.DummyRegressor</code>(strategy, constant, ...)</a>	DummyRegressor is a regressor that makes predictions using simple rules. DummyRegressor - это регрессор, который делает прогнозы, используя простые правила

## **sklearn.ensemble: Ensemble Methods**

The [`sklearn.ensemble`](#) module includes ensemble-based methods for classification, regression and anomaly detection.

Модуль `sklearn.ensemble` включает основанные на ансамбле методы для классификации, регрессии и обнаружения аномалий.

**User guide:** See the [Ensemble methods](#) section for further details.

<a href="#">ensemble.AdaBoostClassifier(...)</a>	An AdaBoost classifier.
<a href="#">ensemble.AdaBoostRegressor([base_estimator, ...])</a>	An AdaBoost regressor.
<a href="#">ensemble.BaggingClassifier([base_estimator, ...])</a>	A Bagging classifier.
<a href="#">ensemble.BaggingRegressor([base_estimator, ...])</a>	A Bagging regressor.
<a href="#">ensemble.ExtraTreesClassifier(...)</a>	An extra-trees classifier.
<a href="#">ensemble.ExtraTreesRegressor([n_estimators, ...])</a>	An extra-trees regressor.
<a href="#">ensemble.GradientBoostingClassifier([loss, ...])</a>	Gradient Boosting for classification. Повышение градиента для классификации.
<a href="#">ensemble.GradientBoostingRegressor([loss, ...])</a>	Gradient Boosting for regression.
<a href="#">ensemble.IsolationForest([n_estimators, ...])</a>	Isolation Forest Algorithm Алгоритм выделения леса
<a href="#">ensemble.RandomForestClassifier(...)</a>	A random forest classifier.
<a href="#">ensemble.RandomForestRegressor(...)</a>	A random forest regressor.
<a href="#">ensemble.RandomTreesEmbedding(...)</a>	An ensemble of totally random trees. Ансамбль совершенно случайных деревьев.
<a href="#">ensemble.VotingClassifier(estimators[, ...])</a>	Soft Voting/Majority Rule classifier for unfitted estimators. Классификатор мягкого голосования / правила большинства для неподходящих оценок.
<a href="#">ensemble.VotingRegressor(estimators[, ...])</a>	Prediction voting regressor for unfitted estimators. Прогнозирование голосования регрессора для неподходящих оценщиков.
<a href="#">ensemble.HistGradientBoostingRegressor(...)</a>	Histogram-based Gradient Boosting Regression Tree. Градиентное дерево регрессии, основанное на гистограмме.
<a href="#">ensemble.HistGradientBoostingClassifier(...)</a>	Histogram-based Gradient Boosting Classification Tree.

## **sklearn.exceptions: Exceptions and warnings**

The [sklearn.exceptions](#) module includes all custom warnings and error classes used across scikit-learn. Модуль sklearn.exceptions включает в себя все пользовательские предупреждения и классы ошибок, используемые в scikit-learn.

<a href="#">exceptions.ChangedBehaviorWarning</a>	Warning class used to notify the user of any change in the behavior. Класс предупреждения, используемый для уведомления пользователя о любых изменениях в поведении.
<a href="#">exceptions.ConvergenceWarning</a>	Custom warning to capture convergence problems Настраиваемое предупреждение для выявления проблем сходимости
<a href="#">exceptions.DataConversionWarning</a>	Warning used to notify implicit data conversions happening in the code. Предупреждение используется для уведомления о неявных преобразованиях данных, происходящих в коде
<a href="#">exceptions.DataDimensionalityWarning</a>	Custom warning to notify potential issues with data dimensionality. Настраиваемое предупреждение для

	уведомления о потенциальных проблемах с размерностью данных.
<a href="#">exceptions.EfficiencyWarning</a>	Warning used to notify the user of inefficient computation. Предупреждение используется для уведомления пользователя о неэффективных вычислениях.
<a href="#">exceptions.FitFailedWarning</a>	Warning class used if there is an error while fitting the estimator. Класс предупреждения используется, если при установке оценщика произошла ошибка.
<a href="#">exceptions.NotFittedError</a>	Exception class to raise if estimator is used before fitting. Класс исключения для повышения, если оценщик используется перед установкой.
<a href="#">exceptions.NonBLASDotWarning</a>	Warning used when the dot operation does not use BLAS. Предупреждение используется, когда операция с точкой не использует BLAS.
<a href="#">exceptions.UndefinedMetricWarning</a>	Warning used when the metric is invalid. Предупреждение используется, когда метрика неверна

## **sklearn.experimental: Experimental**

The [sklearn.experimental](#) module provides importable modules that enable the use of experimental features or estimators.

The features and estimators that are experimental aren't subject to deprecation cycles. Use them at your own risks!

Модуль sklearn.experimental предоставляет импортируемые модули, которые позволяют использовать экспериментальные функции или оценки.

Экспериментальные функции и оценки не подвержены циклам устаревания. Используйте их на свой страх и риск!

<a href="#">experimental.enable_hist_gradient_boosting</a>	Enables histogram-based gradient boosting estimators. Включает основанные на гистограмме оценки повышения градиента.
<a href="#">experimental.enable_iterative_imputer</a>	Enables IterativeImputer. Включает Итеративный Импутер

## **sklearn.feature\_extraction: Feature Extraction**

The [sklearn.feature\\_extraction](#) module deals with feature extraction from raw data. It currently includes methods to extract features from text and images.

Модуль sklearn.feature\_extraction занимается извлечением объектов из необработанных данных. В настоящее время он включает методы для извлечения функций из текста и изображений.

**User guide:** See the [Feature extraction](#) section for further details.

<a href="#">feature_extraction.DictVectorizer([dtype, ...])</a>	Transforms lists of feature-value mappings to vectors. Преобразует списки отображений значения объекта в векторы.
<a href="#">feature_extraction.FeatureHasher([...])</a>	Implements feature hashing, aka the hashing trick. Реализует функцию хеширования, иначе трюк хеширования.

## From images

The [`sklearn.feature\_extraction.image`](#) submodule gathers utilities to extract features from images.

Подмодуль `sklearn.feature_extraction.image` собирает утилиты для извлечения функций из изображений.

<a href="#"><code>feature_extraction.image.extract_patches_2d(...)</code></a>	Reshape a 2D image into a collection of patches преобразовать 2D-изображение в коллекцию патчей
<a href="#"><code>feature_extraction.image.grid_to_graph(n_x, n_y)</code></a>	Graph of the pixel-to-pixel connections График межпиксельных соединений
<a href="#"><code>feature_extraction.image.img_to_graph(img[, ...])</code></a>	Graph of the pixel-to-pixel gradient connections График межпиксельных градиентных соединений
<a href="#"><code>feature_extraction.image.reconstruct_from_patches_2d(...)</code></a>	Reconstruct the image from all of its patches. Восстановите изображение из всех его патчей.
<a href="#"><code>feature_extraction.image.PatchExtractor(...)</code></a>	Extracts patches from a collection of images Извлекает патчи из коллекции изображений

## From text

The [`sklearn.feature\_extraction.text`](#) submodule gathers utilities to build feature vectors from text documents.

Подмодуль `sklearn.feature_extraction.text` собирает утилиты для построения векторов объектов из текстовых документов.

<a href="#"><code>feature_extraction.text.CountVectorizer(...)</code></a>	Convert a collection of text documents to a matrix of token counts Преобразование коллекции текстовых документов в матрицу подсчета токенов
<a href="#"><code>feature_extraction.text.HashingVectorizer(...)</code></a>	Convert a collection of text documents to a matrix of token occurrences Преобразование коллекции текстовых документов в матрицу вхождений токенов
<a href="#"><code>feature_extraction.text.TfidfTransformer(...)</code></a>	Transform a count matrix to a normalized tf or tf-idf representation Преобразовать матрицу подсчета в нормализованное представление tf или tf-idf
<a href="#"><code>feature_extraction.text.TfidfVectorizer(...)</code></a>	Convert a collection of raw documents to a matrix of TF-IDF features. Преобразуйте коллекцию необработанных документов в матрицу функций TF-IDF.

## [`sklearn.feature\_selection`](#): Feature Selection

The [`sklearn.feature\_selection`](#) module implements feature selection algorithms. It currently includes univariate filter selection methods and the recursive feature elimination algorithm.

Модуль `sklearn.feature_selection` реализует алгоритмы выбора функций. В настоящее время он включает одномерные методы выбора фильтров и алгоритм исключения рекурсивных функций.

**User guide:** See the [Feature selection](#) section for further details.

<a href="#"><code>feature_selection.GenericUnivariateSelector</code></a>	Univariate feature selector with configurable strategy.
--	---

<a href="#"><code>t(...)</code></a>	Однофакторный селектор функций с настраиваемой стратегией.
<a href="#"><code>feature_selection.SelectPercentile(...)</code></a>	Select features according to a percentile of the highest scores. Выберите функции в соответствии с процентилем наивысших баллов.
<a href="#"><code>feature_selection.SelectKBest([score_func, k])</code></a>	Select features according to the k highest scores. Выберите функции в соответствии с k самых высоких баллов.
<a href="#"><code>feature_selection.SelectFpr([score_func, alpha])</code></a>	Filter: Select the p-values below alpha based on a FPR test. Фильтр: выберите значения ниже альфа на основе теста FPR.
<a href="#"><code>feature_selection.SelectFdr([score_func, alpha])</code></a>	Filter: Select the p-values for an estimated false discovery rate Фильтр: выберите р-значения для оценки вероятности ложного обнаружения
<a href="#"><code>feature_selection.SelectFromModel(estimator)</code></a>	Meta-transformer for selecting features based on importance weights. Мета-преобразователь для выбора характеристик по весам важности.
<a href="#"><code>feature_selection.SelectFwe([score_func, alpha])</code></a>	Filter: Select the p-values corresponding to Family-wise error rate Фильтр: выберите р-значения, соответствующие частоте ошибок по семейным причинам.
<a href="#"><code>feature_selection.RFE(estimator[, ...])</code></a>	Feature ranking with recursive feature elimination. Ранжирование функций с устранением рекурсивных функций.
<a href="#"><code>feature_selection.RFECV(estimator[, step, ...])</code></a>	Feature ranking with recursive feature elimination and cross-validated selection of the best number of features. Ранжирование функций с рекурсивным удалением функций и перекрестным проверкой выбора лучшего количества функций.
<a href="#"><code>feature_selection.VarianceThreshold([threshold])</code></a>	Feature selector that removes all low-variance features. Селектор функций, который удаляет все функции с низкой дисперсией.
<a href="#"><code>feature_selection.chi2(X, y)</code></a>	Compute chi-squared stats between each non-negative feature and class. Вычислить хи-квадрат характеристики между каждым неотрицательным признаком и классом.
<a href="#"><code>feature_selection.f_classif(X, y)</code></a>	Compute the ANOVA F-value for the provided sample. Вычислить F-значение ANOVA для предоставленного образца.
<a href="#"><code>feature_selection.f_regression(X, y[, center])</code></a>	Univariate linear regression tests. Одномерные линейные регрессионные тесты.
<a href="#"><code>feature_selection.mutual_info_classif(X, y)</code></a>	Estimate mutual information for a discrete target variable. Оцените взаимную информацию для дискретной целевой переменной.
<a href="#"><code>feature_selection.mutual_info_regression(X, y)</code></a>	Estimate mutual information for a continuous target variable. Оцените взаимную информацию для непрерывной целевой переменной.

## **sklearn.gaussian\_process: Gaussian Processes**

The [`sklearn.gaussian\_process`](#) module implements Gaussian Process based regression and classification.

Модуль `sklearn.gaussian_process` реализует регрессию и классификацию на основе гауссовского процесса.

**User guide:** See the [Gaussian Processes](#) section for further details.

## **sklearn.isotonic: Isotonic regression**

**User guide:** See the [Isotonic regression](#) section for further details.

<a href="#"><code>isotonic.IsotonicRegression</code></a> ( <code>ly_min</code> , <code>y_max</code> , ...)	Isotonic regression model.
<a href="#"><code>isotonic.check_increasing</code></a> ( <code>x</code> , <code>y</code> )	Determine whether <code>y</code> is monotonically correlated with <code>x</code> . Определите, монотонно ли коррелирует <code>y</code> с <code>x</code> .
<a href="#"><code>isotonic.isotonic_regression</code></a> ( <code>y</code> [, ...])	Solve the isotonic regression model: Решить модель изотонической регрессии

## **sklearn.impute: Impute**

Transformers for missing value imputation Трансформаторы для пропуска значения вменения

**User guide:** See the [Imputation of missing values](#) section for further details.

<a href="#"><code>impute.SimpleImputer</code></a> ( <code>missing_values</code> , ...)	Imputation transformer for completing missing values. Преобразователь импутации для заполнения пропущенных значений.
<a href="#"><code>impute.IterativeImputer</code></a> ( <code>estimator</code> , ...)	Multivariate imputer that estimates each feature from all the others. Многомерный импутер, который оценивает каждую особенность от всех других.
<a href="#"><code>impute.MissingIndicator</code></a> ( <code>missing_values</code> , ...)	Binary indicators for missing values. Двоичные показатели для пропущенных значений.

## **sklearn.kernel\_approximation Kernel Approximation**

The [`sklearn.kernel\_approximation`](#) module implements several approximate kernel feature maps base on Fourier transforms.

Модуль `sklearn.kernel_approximation` реализует несколько приближенных карт характеристик ядра на основе преобразований Фурье.

**User guide:** See the [Kernel Approximation](#) section for further details.

<a href="#"><code>kernel_approximation.AdditiveChi2Sampler</code></a> ([...])	Approximate feature map for additive chi2 kernel. Примерная карта характеристик для аддитивного ядра chi2.
<a href="#"><code>kernel_approximation.Nystroem</code></a> ( <code>kernel</code> , ...)	Approximate a kernel map using a subset of the training data. Приблизьте карту ядра, используя подмножество обучающих данных.
<a href="#"><code>kernel_approximation.RBFSampler</code></a> ( <code>gamma</code> , ...)	Approximates feature map of an RBF kernel by Monte Carlo approximation of its Fourier transform.



	Аппроксимирует характеристическое отображение ядра RBF приближением Монте-Карло его преобразования Фурье.
<a href="#"><code>kernel_approximation.SkewedChi2Sampler</code></a> ([...])	Approximates feature map of the “skewed chi-squared” kernel by Monte Carlo approximation of its Fourier transform. Аппроксимирует характеристическую карту ядра «скошенного хи-квадрата» приближением Монте-Карло его преобразования Фурье.

## **sklearn.kernel\_ridge Kernel Ridge Regression**

Module [`sklearn.kernel\_ridge`](#) implements kernel ridge regression.

Модуль `sklearn.kernel_ridge` реализует регрессию гребня ядра.

**User guide:** See the [Kernel ridge regression](#) section for further details.

<a href="#"><code>kernel_ridge.KernelRidge</code></a> ([alpha, kernel, ...])	Kernel ridge regression. Регрессия гребня ядра.
--	--

## **sklearn.linear\_model: Generalized Linear Models**

The [`sklearn.linear\_model`](#) module implements generalized linear models. It includes Ridge regression, Bayesian Regression, Lasso and Elastic Net estimators computed with Least Angle Regression and coordinate descent. It also implements Stochastic Gradient Descent related algorithms.

Модуль `sklearn.linear_model` реализует обобщенные линейные модели. Она включает в себя регрессию Риджа, байесовскую регрессию, оценки Лассо и Эластичной сети, рассчитанные с использованием регрессии наименьшего угла и координатного спуска. Он также реализует алгоритмы, связанные со Стохастическим градиентным спуском.

**User guide:** See the [Generalized Linear Models](#) section for further details.

<a href="#"><code>linear_model.ARDRegression</code></a> ([n_iter, tol, ...])	Bayesian ARD regression.
<a href="#"><code>linear_model.BayesianRidge</code></a> ([n_iter, tol, ...])	Bayesian ridge regression.
<a href="#"><code>linear_model.ElasticNet</code></a> ([alpha, l1_ratio, ...])	Linear regression with combined L1 and L2 priors as regularizer. Линейная регрессия с комбинированными априорами L1 и L2 в качестве регуляризатора.
<a href="#"><code>linear_model.ElasticNetCV</code></a> ([l1_ratio, eps, ...])	Elastic Net model with iterative fitting along a regularization path. Модель эластичной сети с итеративной подгонкой вдоль пути регуляризации.
<a href="#"><code>linear_model.HuberRegressor</code></a> ([epsilon, ...])	Linear regression model that is robust to outliers. Модель линейной регрессии, устойчивая к выбросам.
<a href="#"><code>linear_model.Lars</code></a> ([fit_intercept, verbose, ...])	Least Angle Regression model a.k.a. Модель регрессии наименьшего угла a.k.a.
<a href="#"><code>linear_model.LarsCV</code></a> ([fit_intercept, ...])	Cross-validated Least Angle Regression model. Перекрестная модель регрессии наименьшего угла.
<a href="#"><code>linear_model.Lasso</code></a> ([alpha, fit_intercept, ...])	Linear Model trained with L1 prior as regularizer (aka the Lasso) Линейная модель обучена с L1 до регуляризатора (он же Лассо)
<a href="#"><code>linear_model.LassoCV</code></a> ([eps, n_alphas, ...])	Lasso linear model with iterative fitting along a regularization path. Лассо линейная модель с итеративной подгонкой вдоль пути регуляризации.

<a href="#"><code>linear_model.LassoLars(alpha, ...)</code></a>	Lasso model fit with Least Angle Regression a.k.a. Лассо модель подходит с регрессией наименьшего угла a.k.a.
<a href="#"><code>linear_model.LassoLarsCV([fit_intercept, ...])</code></a>	Cross-validated Lasso, using the LARS algorithm. Перекрестная проверка Лассо с использованием алгоритма LARS.
<a href="#"><code>linear_model.LassoLarsIC([criterion, ...])</code></a>	Lasso model fit with Lars using BIC or AIC for model selection Лассо модель подходит с Ларсом, используя BIC или AIC для выбора модели
<a href="#"><code>linear_model.LinearRegression(...)</code></a>	Ordinary least squares Linear Regression. Обычная наименьших квадратов линейная регрессия.
<a href="#"><code>linear_model.LogisticRegression([penalty, ...])</code></a>	Logistic Regression (aka logit, MaxEnt) classifier. Классификатор логистической регрессии (также известный как MaxEnt).
<a href="#"><code>linear_model.LogisticRegressionCV([Cs, ...])</code></a>	Logistic Regression CV (aka logit, MaxEnt) classifier. Классификатор логистической регрессии CV (aka logit, MaxEnt).
<a href="#"><code>linear_model.MultiTaskLasso([alpha, ...])</code></a>	Multi-task Lasso model trained with L1/L2 mixed-norm as regularizer. Многозадачная модель Лассо, обученная со смешанной нормой L1 / L2 в качестве регуляризатора.
<a href="#"><code>linear_model.MultiTaskElasticNet([alpha, ...])</code></a>	Multi-task ElasticNet model trained with L1/L2 mixed-norm as regularizer Многоцелевая модель ElasticNet, обученная со смешанной нормой L1 / L2 в качестве регуляризатора
<a href="#"><code>linear_model.MultiTaskLassoCV([eps, ...])</code></a>	Multi-task Lasso model trained with L1/L2 mixed-norm as regularizer. Многозадачная модель Лассо, обученная со смешанной нормой L1 / L2 в качестве регуляризатора.
<a href="#"><code>linear_model.MultiTaskElasticNetCV(...)</code></a>	Multi-task L1/L2 ElasticNet with built-in cross-validation. Многозадачность L1 / L2 ElasticNet со встроенной перекрестной проверкой
<a href="#"><code>linear_model.OrthogonalMatchingPursuit(...)</code></a>	Orthogonal Matching Pursuit model (OMP) Ортогональная модель преследования
<a href="#"><code>linear_model.OrthogonalMatchingPursuitCV(...)</code></a>	Cross-validated Orthogonal Matching Pursuit model (OMP). Перекрестно подтвержденная модель «Ортогональное соответствие» (OMP).
<a href="#"><code>linear_model.PassiveAggressiveClassifier(...)</code></a>	Passive Aggressive Classifier Пассивный агрессивный классификатор
<a href="#"><code>linear_model.PassiveAggressiveRegressor([C, ...])</code></a>	Passive Aggressive Regressor Пассивный Агрессивный Регрессор
<a href="#"><code>linear_model.Perceptron([penalty, alpha, ...])</code></a>	Read more in the <a href="#">User Guide</a> .
<a href="#"><code>linear_model.RANSACRegressor(...)</code></a>	RANSAC (RANDOM Sample Consensus) algorithm. Случайный, слабый консенсус
<a href="#"><code>linear_model.Ridge([alpha, fit_intercept, ...])</code></a>	Linear least squares with l2 regularization. Линейные наименьшие квадраты с l2 регуляризацией.
<a href="#"><code>linear_model.RidgeClassifier([alpha, ...])</code></a>	Classifier using Ridge regression. Классификатор с использованием хребта регрессии.
<a href="#"><code>linear_model.RidgeClassifierCV([alphas, ...])</code></a>	Ridge classifier with built-in cross-validation. Классификатор гребня со встроенной перекрестной проверкой
<a href="#"><code>linear_model.RidgeCV([alphas, ...])</code></a>	Ridge regression with built-in cross-validation.
<a href="#"><code>linear_model.SGDClassifier([loss, penalty, ...])</code></a>	Linear classifiers (SVM, logistic regression, a.o.) with SGD training.

<a href="#"><code>linear_model.SGDRegressor</code>([loss, penalty, ...])</a>	Linear model fitted by minimizing a regularized empirical loss with SGD Линейная модель подбирается путем минимизации регуляризованных эмпирических потерь с SGD
<a href="#"><code>linear_model.TheilSenRegressor</code>([...])</a>	Theil-Sen Estimator: robust multivariate regression model. Оценщик Тейла-Сен: надежная многомерная регрессионная модель.
<a href="#"><code>linear_model.enet_path</code>(X, y[, l1_ratio, ...])</a>	Compute elastic net path with coordinate descent Вычислить траекторию упругой сетки с координатой
<a href="#"><code>linear_model.lars_path</code>(X, y[, Xy, Gram, ...])</a>	Compute Least Angle Regression or Lasso path using LARS algorithm [1] Вычислить регрессию наименьшего угла или путь Лассо, используя алгоритм LARS [1]
<a href="#"><code>linear_model.lars_path_gram</code>(Xy, Gram, n_samples)</a>	<code>lars_path</code> in the sufficient stats mode [1] <code>lars_path</code> в режиме достаточной статистики [1]
<a href="#"><code>linear_model.lasso_path</code>(X, y[, eps, ...])</a>	Compute Lasso path with coordinate descent Вычислить путь Лассо с координатным спуском
<a href="#"><code>linear_model.orthogonal_mp</code>(X, y[, ...])</a>	Orthogonal Matching Pursuit (OMP) Ортогональное соответствие преследования
<a href="#"><code>linear_model.orthogonal_mp_gram</code>(Gram, Xy[, ...])</a>	Gram Orthogonal Matching Pursuit (OMP) Грамматическое преследование
<a href="#"><code>linear_model.ridge_regression</code>(X, y, alpha[, ...])</a>	Solve the ridge equation by the method of normal equations. Решите уравнение гребня методом нормальных уравнений.

## [\*\*`sklearn.manifold`\*\*](#): Manifold Learning

### Многообразие обучения

The [`sklearn.manifold`](#) module implements data embedding techniques.

Модуль `sklearn.manifold` реализует методы встраивания данных.

**User guide:** See the [Manifold learning](#) section for further details.

<a href="#"><code>manifold.Isomap</code>([n_neighbors, n_components, ...])</a>	Isomap Embedding Встраивание Isomap
<a href="#"><code>manifold.LocallyLinearEmbedding</code>([...])</a>	Locally Linear Embedding Локально линейное вложение
<a href="#"><code>manifold.MDS</code>([n_components, metric, n_init, ...])</a>	Multidimensional scaling Многомерное масштабирование
<a href="#"><code>manifold.SpectralEmbedding</code>([n_components, ...])</a>	Spectral embedding for non-linear dimensionality reduction. Спектральное вложение для нелинейного уменьшения размерности.
<a href="#"><code>manifold.TSNE</code>([n_components, perplexity, ...])</a>	t-distributed Stochastic Neighbor Embedding. t-распределенное стохастическое вложение соседей.

## **sklearn.metrics: Metrics**

See the [Model evaluation: quantifying the quality of predictions](#) section and the [Pairwise metrics, Affinities and Kernels](#) section of the user guide for further details.

The [sklearn.metrics](#) module includes score functions, performance metrics and pairwise metrics and distance computations.

Дополнительные сведения см. В разделе «Оценка модели: количественная оценка качества прогнозов» и «Парные метрики, сходства и ядра» в руководстве пользователя.

Модуль sklearn.metrics включает функции оценки, показатели производительности и попарные показатели, а также вычисления расстояния.

### **Model Selection Interface**

See the [The scoring parameter: defining model evaluation rules](#) section of the user guide for further details.

<a href="#">metrics.check_scoring</a> (estimator[, scoring, ...])	Determine scorer from user options. Определить счетчик из пользовательских опций.
<a href="#">metrics.get_scorer</a> (scoring)	Get a scorer from string Получить счетчик из строки
<a href="#">metrics.make_scorer</a> (score_func[, ...])	Make a scorer from a performance metric or loss function. Сделать счетчик из показателя производительности или функции потерь.

### **Classification metrics**

See the [Classification metrics](#) section of the user guide for further details.

<a href="#">metrics.accuracy_score</a> (y_true, y_pred[, ...])	Accuracy classification score. Точность классификации.
<a href="#">metrics.auc</a> (x, y[, reorder])	Compute Area Under the Curve (AUC) using the trapezoidal rule Вычислить площадь под кривой (AUC) с использованием правила трапеции
<a href="#">metrics.average_precision_score</a> (y_true, y_score)	Compute average precision (AP) from prediction scores Вычислить среднюю точность (AP) на основе результатов прогноза
<a href="#">metrics.balanced_accuracy_score</a> (y_true, y_pred)	Compute the balanced accuracy Вычислить сбалансированную точность
<a href="#">metrics.brier_score_loss</a> (y_true, y_prob[, ...])	Compute the Brier score. Вычислить оценку Бриера.
<a href="#">metrics.classification_report</a> (y_true, y_pred)	Build a text report showing the main classification metrics Создайте текстовый отчет, показывающий основные метрики классификации
<a href="#">metrics.cohen_kappa_score</a> (y1, y2[, labels, ...])	Cohen's kappa: a statistic that measures inter-annotator agreement. Каппа Коэна: статистика, которая измеряет межаннотаторское соглашение.
<a href="#">metrics.confusion_matrix</a> (y_true, y_pred[, ...])	Compute confusion matrix to evaluate the accuracy of a classification Вычислить матрицу путаницы для оценки точности классификации
<a href="#">metrics.f1_score</a> (y_true, y_pred[, labels, ...])	Compute the F1 score, also known as balanced F-score or F-measure Вычислить балл F1, также известный как сбалансированная F-оценка или F-мера
<a href="#">metrics.fbeta_score</a> (y_true, y_pred, beta[, ...])	Compute the F-beta score Вычислить F-бета

<a href="#"><code>metrics.hamming_loss(y_true, y_pred[, ...])</code></a>	Compute the average Hamming loss. Вычислить среднюю потерю Хэмминга.
<a href="#"><code>metrics.hinge_loss(y_true, pred_decision[, ...])</code></a>	Average hinge loss (non-regularized) Средняя потеря шарнира (нерегулируемая)
<a href="#"><code>metrics.jaccard_score(y_true, y_pred[, ...])</code></a>	Jaccard similarity coefficient score Коэффициент коэффициента подобия Жакара
<a href="#"><code>metrics.log_loss(y_true, y_pred[, eps, ...])</code></a>	Log loss, aka logistic loss or cross-entropy loss. Потеря логарифма, то есть логистическая потеря или потеря перекрестной энтропии.
<a href="#"><code>metrics.matthews_corrcoef(y_true, y_pred[, ...])</code></a>	Compute the Matthews correlation coefficient (MCC) Вычислить коэффициент корреляции Мэтьюса
<a href="#"><code>metrics.multilabel_confusion_matrix(y_true, ...)</code></a>	Compute a confusion matrix for each class or sample Вычислить матрицу путаницы для каждого класса или образца
<a href="#"><code>metrics.precision_recall_curve(y_true, ...)</code></a>	Compute precision-recall pairs for different probability thresholds Вычислить пары точности-отзыва для разных порогов вероятности
<a href="#"><code>metrics.precision_recall_fscore_support(...)</code></a>	Compute precision, recall, F-measure and support for each class Вычислите точность, вспомните, F-меру и поддержку для каждого класса
<a href="#"><code>metrics.precision_score(y_true, y_pred[, ...])</code></a>	Compute the precision Вычислить точность
<a href="#"><code>metrics.recall_score(y_true, y_pred[, ...])</code></a>	Compute the recall Вычислить отзыв
<a href="#"><code>metrics.roc_auc_score(y_true, y_score[, ...])</code></a>	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores. Вычислить площадь под кривой рабочих характеристик приемника (ROC AUC) на основе результатов прогнозирования.
<a href="#"><code>metrics.roc_curve(y_true, y_score[, ...])</code></a>	Compute Receiver operating characteristic (ROC) Рабочая характеристика вычислительного приемника
<a href="#"><code>metrics.zero_one_loss(y_true, y_pred[, ...])</code></a>	Zero-one classification loss. Потеря классификации ноль один.

## Regression metrics

See the [Regression metrics](#) section of the user guide for further details.

## Multilabel ranking metrics

See the [Multilabel ranking metrics](#) section of the user guide for further details.

<a href="#">metrics.coverage_error(y_true, y_score[, ...])</a>	Coverage error measure Мера погрешности покрытия
<a href="#">metrics.label_ranking_average_precision_score(...)</a>	Compute ranking-based average precision Вычислить среднюю точность на основе ранжирования
<a href="#">metrics.label_ranking_loss(y_true, y_score)</a>	Compute Ranking loss measure Вычислить рейтинг потери измерения

## Clustering metrics

See the [Clustering performance evaluation](#) section of the user guide for further details.

The [sklearn.metrics.cluster](#) submodule contains evaluation metrics for cluster analysis results. There are two forms of evaluation:

- supervised, which uses a ground truth class values for each sample.
- unsupervised, which does not and measures the ‘quality’ of the model itself.

Подмодуль `sklearn.metrics.cluster` содержит метрики оценки результатов кластерного анализа. Есть две формы оценки:

- супервизированный, который использует базовые значения класса истинности для каждой выборки.
- без присмотра, который не определяет и не измеряет «качество» самой модели.

<a href="#">metrics.adjusted_mutual_info_score(...[, ...])</a>	Adjusted Mutual Information between two clusterings. Скорректированная взаимная информация между двумя кластерами
<a href="#">metrics.adjusted_rand_score(labels_true, ...)</a>	Rand index adjusted for chance. Индекс Рэнда скорректирован на случайность.
<a href="#">metrics.calinski_harabasz_score(X, labels)</a>	Compute the Calinski and Harabasz score. Вычислить баллы Калинского и Харабаса.
<a href="#">metrics.davies_bouldin_score(X, labels)</a>	Computes the Davies-Bouldin score. Вычисляет балл Дэвиса-Болдина.
<a href="#">metrics.completeness_score(labels_true, ...)</a>	Completeness metric of a cluster labeling given a ground truth. Показатель полноты маркировки кластера с учетом истинности оснований.
<a href="#">metrics.cluster_contingency_matrix(...[, ...])</a>	Build a contingency matrix describing the relationship between labels. Создайте матрицу непредвиденных обстоятельств, описывающую отношения между метками.
<a href="#">metrics.fowlkes_mallows_score(labels_true, ...)</a>	Measure the similarity of two clusterings of a set of points. Измерьте сходство двух кластеров набора точек.
<a href="#">metrics.homogeneity_completeness_v_measure(...)</a>	Compute the homogeneity and completeness and V-Measure scores at once. Вычислить однородность и полноту и баллы V-меры сразу.
<a href="#">metrics.homogeneity_score(labels_true, ...)</a>	Homogeneity metric of a cluster labeling given a ground truth. Метрика однородности маркировки кластера дает основную правду.
<a href="#">metrics.mutual_info_score(labels_true, ...)</a>	Mutual Information between two clusterings. Взаимная информация между двумя кластерами.

<a href="#"><code>metrics.normalized_mutual_info_score(...[, ...])</code></a>	Normalized Mutual Information between two clusterings. Нормализованная взаимная информация между двумя кластерами.
<a href="#"><code>metrics.silhouette_score(X, labels[, ...])</code></a>	Compute the mean Silhouette Coefficient of all samples. Вычислите средний Коэффициент Силуэт всех образцов.
<a href="#"><code>metrics.silhouette_samples(X, labels[, metric])</code></a>	Compute the Silhouette Coefficient for each sample. Вычислить Коэффициент Силуэт для каждого образца.
<a href="#"><code>metrics.v_measure_score(labels_true, labels_pred)</code></a>	V-measure cluster labeling given a ground truth. Маркировка кластера V-меры дает основную правду.

## Biclustering metrics

See the [Biclustering evaluation](#) section of the user guide for further details.

<a href="#"><code>metrics.consensus_score(a, b[, similarity])</code></a>	The similarity of two sets of biclusters. Сходство двух наборов бикластеров.
--	--

## Pairwise metrics

See the [Pairwise metrics, Affinities and Kernels](#) section of the user guide for further details.

<a href="#"><code>metrics.pairwise.additive_chi2_kernel(X[, Y])</code></a>	Computes the additive chi-squared kernel between observations in X and Y. Вычисляет аддитивное ядро хи-квадрат между наблюдениями в X и Y
<a href="#"><code>metrics.pairwise.chi2_kernel(X[, Y, gamma])</code></a>	Computes the exponential chi-squared kernel X and Y. Вычисляет экспоненциальное ядро хи-квадрат X и Y.
<a href="#"><code>metrics.pairwise.cosine_similarity(X[, Y, ...])</code></a>	Compute cosine similarity between samples in X and Y. Вычислить косинусное сходство между образцами в X и Y
<a href="#"><code>metrics.pairwise.cosine_distances(X[, Y])</code></a>	Compute cosine distance between samples in X and Y. Вычислить косинусное расстояние между образцами в X и Y.
<a href="#"><code>metrics.pairwise.distance_metrics()</code></a>	Valid metrics for pairwise_distances. Допустимые метрики для pairwise_distances.
<a href="#"><code>metrics.pairwise.euclidean_distances(X[, Y, ...])</code></a>	Considering the rows of X (and Y=X) as vectors, compute the distance matrix between each pair of vectors. Рассматривая строки X (и Y = X) как векторы, вычислите матрицу расстояний между каждой парой векторов.
<a href="#"><code>metrics.pairwise.haversine_distances(X[, Y])</code></a>	Compute the Haversine distance between samples in X and Y. Вычислить расстояние Хаверсайна между выборками в X и Y
<a href="#"><code>metrics.pairwise.kernel_metrics()</code></a>	Valid metrics for pairwise_kernels. Вычислить расстояние Хаверсайна между выборками в X и Y
<a href="#"><code>metrics.pairwise.laplacian_kernel(X[, Y, gamma])</code></a>	Compute the laplacian kernel between X and Y. Вычислить ядро лапласиана между X и Y.
<a href="#"><code>metrics.pairwise.linear_kernel(X[, Y, ...])</code></a>	Compute the linear kernel between X and Y. Вычислить линейное ядро между X и Y.
<a href="#"><code>metrics.pairwise.manhattan_distances(X[, Y, ...])</code></a>	Compute the L1 distances between the vectors in X and Y. Вычислить расстояния L1 между векторами в X и Y.



<a href="#"><code>metrics.pairwise.pairwise_kernels(X[, Y, ...])</code></a>	Compute the kernel between arrays X and optional array Y. Вычислить ядро между массивами X и необязательным массивом Y.
<a href="#"><code>metrics.pairwise.polynomial_kernel(X[, Y, ...])</code></a>	Compute the polynomial kernel between X and Y: Вычислить ядро полинома между X и Y:
<a href="#"><code>metrics.pairwise.rbf_kernel(X[, Y, gamma])</code></a>	Compute the rbf (gaussian) kernel between X and Y: Вычислить ядро rbf (гауссово) между X и Y:
<a href="#"><code>metrics.pairwise.sigmoid_kernel(X[, Y, ...])</code></a>	Compute the sigmoid kernel between X and Y: Вычислить сигмовидное ядро между X и Y:
<a href="#"><code>metrics.pairwise.paired_euclidean_distances(X, Y)</code></a>	Computes the paired euclidean distances between X and Y. Вычисляет парные евклидовы расстояния между X и Y
<a href="#"><code>metrics.pairwise.paired_manhattan_distances(X, Y)</code></a>	Compute the L1 distances between the vectors in X and Y. Вычислить расстояния L1 между векторами в X и Y.
<a href="#"><code>metrics.pairwise.paired_cosine_distances(X, Y)</code></a>	Computes the paired cosine distances between X and Y. Вычисляет парные косинусные расстояния между X и Y
<a href="#"><code>metrics.pairwise.paired_distances(X, Y[, metric])</code></a>	Computes the paired distances between X and Y. Вычисляет парные расстояния между X и Y.
<a href="#"><code>metrics.pairwise_distances(X[, Y, metric, ...])</code></a>	Compute the distance matrix from a vector array X and optional Y. Вычислить матрицу расстояний из векторного массива X и необязательного Y.
<a href="#"><code>metrics.pairwise_distances_argmin(X, Y[, ...])</code></a>	Compute minimum distances between one point and a set of points. Вычислить минимальные расстояния между одной точкой и набором точек.
<a href="#"><code>metrics.pairwise_distances_argmin_min(X, Y)</code></a>	Compute minimum distances between one point and a set of points. Вычислить минимальные расстояния между одной точкой и набором точек.
<a href="#"><code>metrics.pairwise_distances_chunked(X[, Y, ...])</code></a>	Generate a distance matrix chunk by chunk with optional reduction Создание фрагмента матрицы расстояний по фрагменту с необязательным сокращением

## **sklearn.mixture: Gaussian Mixture Models**

The [`sklearn.mixture`](#) module implements mixture modeling algorithms.

Модуль `sklearn.mixture` реализует алгоритмы моделирования смеси.

**User guide:** See the [Gaussian mixture models](#) section for further details.

<a href="#"><code>mixture.BayesianGaussianMixture(...)</code></a>	Variational Bayesian estimation of a Gaussian mixture. Вариационная байесовская оценка гауссовой смеси.
<a href="#"><code>mixture.GaussianMixture(n_components, ...)</code></a>	Gaussian Mixture. Гауссова смесь.

## **sklearn.model\_selection: Model Selection**

**User guide:** See the [Cross-validation: evaluating estimator performance](#), [Tuning the hyper-parameters of an estimator](#) and [Learning curve](#) sections for further details.

## Splitter Classes

<a href="#"><code>model_selection.GroupKFold([n_splits])</code></a>	K-fold iterator variant with non-overlapping groups. К-кратный вариант итератора с неперекрывающимися группами.
<a href="#"><code>model_selection.GroupShuffleSplit(...)</code></a>	Shuffle-Group(s)-Out cross-validation iterator Shuffle-Group (s) -Out итератор перекрестной проверки
<a href="#"><code>model_selection.KFold([n_splits, shuffle, ...])</code></a>	K-Folds cross-validator Перекрестная проверка K-Folds
<a href="#"><code>model_selection.LeaveOneGroupOut</code></a>	Leave One Group Out cross-validator Оставьте одну группу вне перекрестной проверки
<a href="#"><code>model_selection.LeavePGroupsOut(n_groups)</code></a>	Leave P Group(s) Out cross-validator Оставьте P Group (s) вне перекрестной проверки
<a href="#"><code>model_selection.LeaveOneOut</code></a>	Leave-One-Out cross-validator Перекрестная проверка по принципу «оставь один раз»
<a href="#"><code>model_selection.LeavePOut(p)</code></a>	Leave-P-Out cross-validator Перекрестная проверка Leave-P-Out
<a href="#"><code>model_selection.PredefinedSplit(test_fold)</code></a>	Predefined split cross-validator Предопределенный разделенный перекрестный валидатор
<a href="#"><code>model_selection.RepeatedKFold([n_splits, ...])</code></a>	Repeated K-Fold cross validator. Повторная перекрестная проверка K-Fold.
<a href="#"><code>model_selection.RepeatedStratifiedKFold(...)</code></a>	Repeated Stratified K-Fold cross validator. Повторная многоуровневая перекрестная проверка K-Fold.
<a href="#"><code>model_selection.ShuffleSplit([n_splits, ...])</code></a>	Random permutation cross-validator Кросс-валидатор случайной перестановки
<a href="#"><code>model_selection.StratifiedKFold([n_splits, ...])</code></a>	Stratified K-Folds cross-validator Стратифицированная перекрестная проверка K-Folds
<a href="#"><code>model_selection.StratifiedShuffleSplit(...)</code></a>	Stratified ShuffleSplit cross-validator Перекрестная проверка со стратифицированным перемешиванием
<a href="#"><code>model_selection.TimeSeriesSplit([n_splits, ...])</code></a>	Time Series cross-validator Перекрестная проверка временных рядов

## Splitter Functions

<a href="#"><code>Model_selection.check_cv([cv, y, classifier])</code></a>	Input checker utility for building a cross-validator Утилита проверки ввода для построения кросс-валидатора
<a href="#"><code>model_selection.train_test_split(*arrays, ...)</code></a>	Split arrays or matrices into random train and test subsets Разбить массивы или матрицы на случайные наборы поездов и тестов

## Hyper-parameter optimizers

<a href="#"><code>model_selection.GridSearchCV(estimator, ...)</code></a>	Exhaustive search over specified parameter values for an estimator. Исчерпывающий поиск по указанным значениям параметров для оценщика.
<a href="#"><code>model_selection.ParameterGrid(param_grid)</code></a>	Grid of parameters with a discrete number of values for each. Сетка параметров с дискретным количеством значений для каждого.
<a href="#"><code>model_selection.ParameterSampler(...[, ...])</code></a>	Generator on parameters sampled from given distributions. Генератор по параметрам, выбранным из заданных распределений.
<a href="#"><code>model_selection.RandomizedSearchCV(...[, ...])</code></a>	Randomized search on hyper parameters.

)	Рандомизированный поиск по гиперпараметрам.
<a href="#">model_selection.fit_grid_point(X, y, ..., ...)</a>	Run fit on one set of parameters. Выполнить посадку по одному набору параметров.

## Model validation

<a href="#">model_selection.cross_validate(estimator, X)</a>	Evaluate metric(s) by cross-validation and also record fit/score times. Оцените метрики с помощью перекрестной проверки, а также запишите время подгонки / оценки.
<a href="#">model_selection.cross_val_predict(estimator, X)</a>	Generate cross-validated estimates for each input data point Генерация перекрестно проверенных оценок для каждой точки входных данных
<a href="#">model_selection.cross_val_score(estimator, X)</a>	Evaluate a score by cross-validation Оценить счет путем перекрестной проверки
<a href="#">model_selection.learning_curve(estimator, X, y)</a>	Learning curve. Кривая обучения.
<a href="#">model_selection.permutation_test_score(...)</a>	Evaluate the significance of a cross-validated score with permutations Оцените значимость перекрестно проверенной оценки с перестановками
<a href="#">model_selection.validation_curve(estimator, ...)</a>	Validation curve. Кривая валидации.

## [sklearn.multiclass](#): Multiclass and multilabel classification

### Multiclass and multilabel classification strategies

This module implements multiclass learning algorithms:

- one-vs-the-rest / one-vs-all
- one-vs-one
- error correcting output codes

The estimators provided in this module are meta-estimators: they require a base estimator to be provided in their constructor. For example, it is possible to use these estimators to turn a binary classifier or a regressor into a multiclass classifier. It is also possible to use these estimators with multiclass estimators in the hope that their accuracy or runtime performance improves.

All classifiers in scikit-learn implement multiclass classification; you only need to use this module if you want to experiment with custom multiclass strategies.

The one-vs-the-rest meta-classifier also implements a [predict\\_proba](#) method, so long as such a method is implemented by the base classifier. This method returns probabilities of class membership in both the single label and multilabel case. Note that in the multilabel case, probabilities are the marginal probability that a given sample falls in the given class. As such, in the multilabel case the sum of these probabilities over all possible labels for a given sample *will not* sum to unity, as they do in the single label case.

Многоклассовая и мультиклассовая классификация стратегий

Этот модуль реализует мультиклассовые алгоритмы обучения:

- один против всех / один против всех

- один против одного
- исправление ошибок выходных кодов

Оценщики, представленные в этом модуле, являются метаоценщиками: они требуют, чтобы базовый оценщик был предоставлен в их конструкторе. Например, можно использовать эти оценки, чтобы превратить двоичный классификатор или регрессор в классификатор мультикласса. Также возможно использовать эти оценщики с мультиклассовыми оценщиками в надежде, что их точность или производительность во время выполнения улучшатся. Все классификаторы в Scikit-Learn реализуют мультиклассовую классификацию; Вам нужно использовать этот модуль, только если вы хотите поэкспериментировать с пользовательскими мультиклассовыми стратегиями.

Мета-классификатор «один против остальных» также реализует метод предиката\_процента, если такой метод реализуется базовым классификатором. Этот метод возвращает вероятности членства в классе как в случае одной метки, так и в случае нескольких меток. Обратите внимание, что в случае с несколькими метками вероятности - это предельная вероятность того, что данная выборка попадает в данный класс. Таким образом, в случае с несколькими метками сумма этих вероятностей по всем возможным меткам для данного образца не будет равна единице, как в случае с одной меткой.

**User guide:** See the [Multiclass and multilabel algorithms](#) section for further details.

<a href="#">multiclass.OneVsRestClassifier</a> (estimator[, ...])	One-vs-the-rest (OvR) multiclass/multilabel strategy Мультиклассовая / мультиметочная стратегия «один против остальных» (OvR)
<a href="#">multiclass.OneVsOneClassifier</a> (estimator[, ...])	One-vs-one multiclass strategy Мультиклассовая стратегия «один против одного»
<a href="#">multiclass.OutputCodeClassifier</a> (estimator[, ...])	(Error-Correcting) Output-Code multiclass strategy (Исправление ошибок) Выходной код мультиклассовой стратегии

## **sklearn.multioutput: Multioutput regression and classification**

This module implements multioutput regression and classification.

The estimators provided in this module are meta-estimators: they require a base estimator to be provided in their constructor. The meta-estimator extends single output estimators to multioutput estimators.

Этот модуль реализует многопотоковую регрессию и классификацию.

Оценщики, представленные в этом модуле, являются метаоценщиками: они требуют, чтобы базовый оценщик был предоставлен в их конструкторе. Метаоценщик распространяет единичные выходные оценки на множественные выходные оценки.

**User guide:** See the [Multiclass and multilabel algorithms](#) section for further details.

<a href="#">multioutput.ClassifierChain</a> (base_estimator)	A multi-label model that arranges binary classifiers into a chain. Модель с несколькими метками, которая объединяет двоичные классификаторы в цепочку.
<a href="#">multioutput.MultiOutputRegressor</a> (estimator)	Multi target regression Многоцелевая регрессия

<a href="#"><code>multioutput.MultiOutputClassifier</code></a> (estimator)	Multi target classification Многоцелевая классификация
<a href="#"><code>multioutput.RegressorChain</code></a> (base_estimator[, ...])	A multi-label model that arranges regressions into a chain. Модель с несколькими метками, которая организует регрессии в цепочку.

## [sklearn.naive\\_bayes](#): Naive Bayes

The [sklearn.naive\\_bayes](#) module implements Naive Bayes algorithms. These are supervised learning methods based on applying Bayes' theorem with strong (naive) feature independence assumptions.

Модуль `sklearn.naive_bayes` реализует наивные байесовские алгоритмы. Это контролируемые методы обучения, основанные на применении теоремы Байеса с сильными (наивными) предположениями о независимости признаков.

**User guide:** See the [Naive Bayes](#) section for further details.

<a href="#"><code>naive_bayes.BernoulliNB</code></a> ([alpha, binarize, ...])	Naive Bayes classifier for multivariate Bernoulli models. Наивный байесовский классификатор для многомерных моделей Бернулли.
<a href="#"><code>naive_bayes.GaussianNB</code></a> ([priors, var_smoothing])	Gaussian Naive Bayes (GaussianNB) Гауссовский Наивный Байес (GaussianNB)
<a href="#"><code>naive_bayes.MultinomialNB</code></a> ([alpha, ...])	Naive Bayes classifier for multinomial models. Наивный байесовский классификатор для полиномиальных моделей
<a href="#"><code>naive_bayes.ComplementNB</code></a> ([alpha, fit_prior, ...])	The Complement Naive Bayes classifier described in Rennie et al. Наивный байесовский классификатор Complement описан в Rennie et al.

## [sklearn.neighbors](#): Nearest Neighbors

The [sklearn.neighbors](#) module implements the k-nearest neighbors algorithm.

Модуль `sklearn.neighbors` реализует алгоритм k-ближайших соседей.

**User guide:** See the [Nearest Neighbors](#) section for further details.

<a href="#"><code>neighbors.BallTree</code></a>	BallTree for fast generalized N-point problems BallTree для быстрых обобщенных N-точечных задач
<a href="#"><code>neighbors.DistanceMetric</code></a>	DistanceMetric class РасстояниеМетрический класс
<a href="#"><code>neighbors.KDTree</code></a>	KDTree for fast generalized N-point problems KDTree для быстрых обобщенных N-точечных задач
<a href="#"><code>neighbors.KernelDensity</code></a> ([bandwidth, ...])	Kernel Density Estimation Оценка плотности ядра
<a href="#"><code>neighbors.KNeighborsClassifier</code></a> ([...])	Classifier implementing the k-nearest neighbors vote. Классификатор, реализующий голосование ближайших соседей.
<a href="#"><code>neighbors.KNeighborsRegressor</code></a> ([n_neighbors, ...])	Regression based on k-nearest neighbors. Регрессия на основе k-ближайших соседей.
<a href="#"><code>neighbors.LocalOutlierFactor</code></a> ([n_neighbors, ...])	Unsupervised Outlier Detection using Local Outlier Factor (LOF) Обнаружение неконтролируемых выбросов с использованием локального фактора выбросов (LOF)

<a href="#"><code>neighbors.RadiusNeighborsClassifier(...)</code></a>	Classifier implementing a vote among neighbors within a given radius. Классификатор, реализующий голосование среди соседей в заданном радиусе
<a href="#"><code>neighbors.RadiusNeighborsRegressor([radius, ...])</code></a>	Regression based on neighbors within a fixed radius. Регрессия на основе соседей в пределах фиксированного радиуса.
<a href="#"><code>neighbors.NearestCentroid([metric, ...])</code></a>	Nearest centroid classifier. Ближайший центроидный классификатор.
<a href="#"><code>neighbors.NearestNeighbors([n_neighbors, ...])</code></a>	Unsupervised learner for implementing neighbor searches. Необучаемый ученик для осуществления поиска соседей.
<a href="#"><code>neighbors.NeighborhoodComponentsAnalysis([...])</code></a>	Neighborhood Components Analysis. Анализ компонентов соседства
<a href="#"><code>neighbors.kneighbors_graph(X, n_neighbors[, ...])</code></a>	Computes the (weighted) graph of k-Neighbors for points in X. Вычисляет (взвешенный) граф k-соседей для точек в X
<a href="#"><code>neighbors.radius_neighbors_graph(X, radius)</code></a>	Computes the (weighted) graph of Neighbors for points in X. Вычисляет (взвешенный) граф соседей для точек в X

## [\*\*sklearn.neural\\_network: Neural network models\*\*](#)

The [`sklearn.neural\_network`](#) module includes models based on neural networks.

Модуль `sklearn.neural_network` включает модели, основанные на нейронных сетях.

**User guide:** See the [Neural network models \(supervised\)](#) and [Neural network models \(unsupervised\)](#) sections for further details.

<a href="#"><code>neural_network.BernoulliRBM([n_components, ...])</code></a>	Bernoulli Restricted Boltzmann Machine (RBM).
<a href="#"><code>neural_network.MLPClassifier(...)</code></a>	Multi-layer Perceptron classifier.
<a href="#"><code>neural_network.MLPRegressor(...)</code></a>	Multi-layer Perceptron regressor.

## [\*\*sklearn.pipeline: Pipeline\*\*](#)

The [`sklearn.pipeline`](#) module implements utilities to build a composite estimator, as a chain of transforms and estimators.

Модуль `sklearn.pipeline` реализует утилиты для построения составной оценки в виде цепочки преобразований и оценок.

<a href="#"><code>pipeline.FeatureUnion(transformer_list[, ...])</code></a>	Concatenates results of multiple transformer objects. Объединяет результаты нескольких объектов трансформатора.
<a href="#"><code>pipeline.Pipeline(steps[, memory, verbose])</code></a>	Pipeline of transforms with a final estimator. Трубопровод преобразований с окончательной оценкой.
<a href="#"><code>pipeline.make_pipeline(*steps, **kwargs)</code></a>	Construct a Pipeline from the given estimators. Построить трубопровод из данных оценок.
<a href="#"><code>pipeline.make_union(*transformers, **kwargs)</code></a>	Construct a FeatureUnion from the given transformers. Построить FeatureUnion из заданных трансформаторов.

## sklearn.inspection: inspection

The [sklearn.inspection](#) module includes tools for model inspection.

Модуль sklearn.inspection включает в себя инструменты для проверки моделей.

<a href="#">inspection.partial_dependence</a> (estimator, X, ...)	Partial dependence of features. Частичная зависимость особенностей.
<a href="#">inspection.plot_partial_dependence</a> (..., ...)	Partial dependence plots. Частичные зависимости участков.

## sklearn.preprocessing: Preprocessing and Normalization

The [sklearn.preprocessing](#) module includes scaling, centering, normalization, binarization and imputation methods.

Модуль sklearn.preprocessing включает методы масштабирования, центрирования, нормализации, бинаризации и вменения.

**User guide:** See the [Preprocessing data](#) section for further details.

<a href="#">Preprocessing.Binarizer</a> ([threshold, copy])	Binarize data (set feature values to 0 or 1) according to a threshold Бинаризация данных (установите значения объектов на 0 или 1) в соответствии с порогом
<a href="#">preprocessing.FunctionTransformer</a> ([func, ...])	Constructs a transformer from an arbitrary callable. Создает преобразователь из произвольного вызова.
<a href="#">preprocessing.KBinsDiscretizer</a> ([n_bins, ...])	Bin continuous data into intervals. Складывайте непрерывные данные в интервалы.
<a href="#">preprocessing.KernelCenterer</a> ()	Center a kernel matrix Центрировать ядро матрицы
<a href="#">preprocessing.LabelBinarizer</a> ([neg_label, ...])	Binarize labels in a one-vs-all fashion Бинаризируйте этикетки в едином стиле
<a href="#">preprocessing.LabelEncoder</a>	Encode labels with value between 0 and n_classes-1. Кодировать метки со значением от 0 до n_classes-1.
<a href="#">preprocessing.MultiLabelBinarizer</a> ([classes, ...])	Transform between iterable of iterables and a multilabel format Преобразование между итерируемыми из итераций в многослойный формат
<a href="#">preprocessing.MaxAbsScaler</a> ([copy])	Scale each feature by its maximum absolute value. Масштабируйте каждую функцию по ее максимальному абсолютному значению.
<a href="#">preprocessing.MinMaxScaler</a> ([feature_range, copy])	Transforms features by scaling each feature to a given range. Преобразует объекты путем масштабирования каждого объекта до заданного диапазона.
<a href="#">preprocessing.Normalizer</a> ([norm, copy])	Normalize samples individually to unit norm. Нормализуйте образцы индивидуально до единицы нормы
<a href="#">preprocessing.OneHotEncoder</a> ([n_values, ...])	Encode categorical integer features as a one-hot numeric array. Кодировать категориальные целочисленные функции в виде однозначного числового массива.
<a href="#">preprocessing.OrdinalEncoder</a> ([categories, dtype])	Encode categorical features as an integer array.



<code>)</code>	Кодировать категориальные признаки в виде целочисленного массива.
<code>preprocessing.PolynomialFeatures([degree, ...])</code>	Generate polynomial and interaction features. Генерация функций полинома и взаимодействия.
<code>preprocessing.PowerTransformer([method, ...])</code>	Apply a power transform featurewise to make data more Gaussian-like. Примените степенное преобразование, чтобы сделать данные более похожими на гауссову.
<code>preprocessing.QuantileTransformer(...)</code>	Transform features using quantiles information. Преобразуйте объекты, используя информацию о квантилях.
<code>preprocessing.RobustScaler([with_centering, ...])</code>	Scale features using statistics that are robust to outliers. Масштабирование объектов с использованием статистики, устойчивой к выбросам.
<code>preprocessing.StandardScaler([copy, ...])</code>	Standardize features by removing the mean and scaling to unit variance Стандартизируйте функции путем удаления среднего значения и масштабирования до дисперсии единиц
<code>preprocessing.add_dummy_feature(X[, value])</code>	Augment dataset with an additional dummy feature.
<code>preprocessing.binarize(X[, threshold, copy])</code>	Boolean thresholding of array-like or scipy.sparse matrix
<code>preprocessing.label_binarize(y, classes[, ...])</code>	Binarize labels in a one-vs-all fashion
<code>preprocessing.maxabs_scale(X[, axis, copy])</code>	Scale each feature to the [-1, 1] range without breaking the sparsity.
<code>preprocessing.minmax_scale(X[, ...])</code>	Transforms features by scaling each feature to a given range. Преобразует объекты путем масштабирования каждого объекта до заданного диапазона.
<code>preprocessing.normalize(X[, norm, axis, ...])</code>	Scale input vectors individually to unit norm (vector length). Масштабирование входных векторов индивидуально до единичной нормы (длины вектора).
<code>preprocessing.quantile_transform(X[, axis, ...])</code>	Transform features using quantiles information. Преобразуйте объекты, используя информацию о квантилях.
<code>preprocessing.robust_scale(X[, axis, ...])</code>	Standardize a dataset along any axis Стандартизировать набор данных по любой оси
<code>preprocessing.scale(X[, axis, with_mean, ...])</code>	Standardize a dataset along any axis Стандартизировать набор данных по любой оси
<code>preprocessing.power_transform(X[, method, ...])</code>	Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. Степенные преобразования - это семейство параметрических, монотонных преобразований, которые применяются для того, чтобы сделать данные более похожими на гауссову.

## sklearn.random\_projection: Random projection

Random Projection transformers

Random Projections are a simple and computationally efficient way to reduce the dimensionality of the data by trading a controlled amount of accuracy (as additional variance) for faster processing times and smaller model sizes.

The dimensions and distribution of Random Projections matrices are controlled so as to preserve the pairwise distances between any two samples of the dataset.

The main theoretical result behind the efficiency of random projection is the [Johnson-Lindenstrauss lemma \(quoting Wikipedia\)](#):

In mathematics, the Johnson-Lindenstrauss lemma is a result concerning low-distortion embeddings of points from high-dimensional into low-dimensional Euclidean space. The lemma states that a small set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. The map used for the embedding is at least Lipschitz, and can even be taken to be an orthogonal projection.

Трансформаторы с произвольной проекцией

Случайные проекции - это простой и эффективный в вычислительном отношении способ уменьшить размерность данных, торгуя контролируемой величиной точности (как дополнительная дисперсия) для более быстрого времени обработки и меньших размеров модели.

Размеры и распределение матриц случайных проекций управляются так, чтобы сохранить попарные расстояния между любыми двумя выборками набора данных. Основным теоретическим результатом эффективности случайной проекции является лемма Джонсона-Линденштраусса (цитата из Википедии):

В математике лемма Джонсона-Линденштраусса является результатом, касающимся вложений точек с низким искажением из многомерного в низкоразмерное евклидово пространство. Лемма утверждает, что небольшой набор точек в многомерном пространстве может быть встроен в пространство гораздо меньшего размера таким образом, что расстояния между точками почти сохраняются. Карта, используемая для вложения, является, по крайней мере, липшицевой, и ее даже можно считать ортогональной проекцией.

**User guide:** See the [Random Projection](#) section for further details.

## **sklearn.semi\_supervised Semi-Supervised Learning**

The [sklearn.semi\\_supervised](#) module implements semi-supervised learning algorithms. These algorithms utilized small amounts of labeled data and large amounts of unlabeled data for classification tasks. This module includes Label Propagation.

Модуль sklearn.semi\_supervised реализует полуобучаемые алгоритмы обучения. Эти алгоритмы использовали небольшие объемы помеченных данных и большие объемы немеченых данных для задач классификации. Этот модуль включает распространение метки.

**User guide:** See the [Semi-Supervised](#) section for further details.

<a href="#">sklearn.semi_supervised.LabelPropagation</a> ([kernel, ...])	Label Propagation classifier Классификатор распространения метки
<a href="#">sklearn.semi_supervised.LabelSpreading</a> ([kernel, ...])	LabelSpreading model for semi-supervised learning Модель LabelSpreading для обучения под наблюдением

## **sklearn.svm: Support Vector Machines**

The [`sklearn.svm`](#) module includes Support Vector Machine algorithms.

Модуль `sklearn.svm` включает в себя алгоритмы опорных векторов.

**User guide:** See the [Support Vector Machines](#) section for further details.

### **Estimators**

<a href="#"><code>Svm.LinearSVC</code></a> ([penalty, loss, dual, tol, C, ...])	Linear Support Vector Classification. Классификация линейных опорных векторов.
<a href="#"><code>svm.LinearSVR</code></a> ([epsilon, tol, C, loss, ...])	Linear Support Vector Regression. Линейная опорная векторная регрессия.
<a href="#"><code>svm.NuSVC</code></a> ([nu, kernel, degree, gamma, ...])	Nu-Support Vector Classification. Классификация векторов Nu-Support.
<a href="#"><code>svm.NuSVR</code></a> ([nu, C, kernel, degree, gamma, ...])	Nu Support Vector Regression. Ню-опорная векторная регрессия.
<a href="#"><code>svm.OneClassSVM</code></a> ([kernel, degree, gamma, ...])	Unsupervised Outlier Detection. Обнаружение неконтролируемых выбросов.
<a href="#"><code>svm.SVC</code></a> ([C, kernel, degree, gamma, coef0, ...])	C-Support Vector Classification. C-опорная векторная классификация.
<a href="#"><code>svm.SVR</code></a> ([kernel, degree, gamma, coef0, tol, ...])	Epsilon-Support Vector Regression. Эпсилон-опорная векторная регрессия.
<a href="#"><code>svm.l1_min_c</code></a> (X, y[, loss, fit_intercept, ...])	Return the lowest bound for C such that for C in (l1_min_C, infinity) the model is guaranteed not to be empty. Верните нижнюю границу для C, так что для C в (l1_min_C, infinity) модель гарантированно не будет пустой.

### **Low-level methods**

<a href="#"><code>svm.libsvm.cross_validation</code></a> ()	Binding of the cross-validation routine (low-level routine) Связывание подпрограммы перекрестной проверки (подпрограмма низкого уровня)
<a href="#"><code>svm.libsvm.decision_function</code></a> ()	Predict margin (libsvm name for this is predict_values) Предсказание полей (имя libsvm для этого — предикат_значения)
<a href="#"><code>svm.libsvm.fit</code></a> ()	Train the model using libsvm (low-level method) Тренируйте модель, используя libsvm (метод низкого уровня)
<a href="#"><code>svm.libsvm.predict</code></a> ()	Predict target values of X given a model (low-level method) Прогнозировать целевые значения X для данной модели (низкоуровневый метод)
<a href="#"><code>svm.libsvm.predict_proba</code></a> ()	Predict probabilities Предсказать вероятности

## **sklearn.tree: Decision Trees**

The [`sklearn.tree`](#) module includes decision tree-based models for classification and regression.

Модуль `sklearn.tree` включает модели дерева решений для классификации и регрессии.

**User guide:** See the [Decision Trees](#) section for further details.

<a href="#"><code>tree.DecisionTreeClassifier</code></a> ([criterion, ...])	A decision tree classifier. Классификатор дерева решений.
---	--

<a href="#"><code>tree.DecisionTreeRegressor</code></a> ([criterion, ...])	A decision tree regressor. Регрессор дерева решений.
<a href="#"><code>tree.ExtraTreeClassifier</code></a> ([criterion, ...])	An extremely randomized tree classifier. Очень рандомизированный древовидный классификатор.
<a href="#"><code>tree.ExtraTreeRegressor</code></a> ([criterion, ...])	An extremely randomized tree regressor. Чрезвычайно рандомизированный регрессор деревьев.
<a href="#"><code>tree.export_graphviz</code></a> (decision_tree[, ...])	Export a decision tree in DOT format. Экспорт дерева решений в формате DOT.
<a href="#"><code>tree.plot_tree</code></a> (decision_tree[, max_depth, ...])	Plot a decision tree. Построить дерево решений.
<a href="#"><code>tree.export_text</code></a> (decision_tree[, ...])	Build a text report showing the rules of a decision tree. Создайте текстовый отчет, показывающий правила дерева решений.

## [\*\*sklearn.utils: Utilities\*\*](#)

The [`sklearn.utils`](#) module includes various utilities.

Модуль `sklearn.utils` включает в себя различные утилиты.

**Developer guide:** See the [Utilities for Developers](#) page for further details.

<a href="#"><code>utils.arrayfuncs.cholesky_delete</code></a> (L, go_out)	
<a href="#"><code>utils.arrayfuncs.min_pos</code></a> ()	Find the minimum value of an array over positive values Найти минимальное значение массива по положительным значениям
<a href="#"><code>utils.as_float_array</code></a> (X[, copy, force_all_finite])	Converts an array-like to an array of floats. Преобразует массив, похожий на массив с плавающей точкой.
<a href="#"><code>utils.assert_all_finite</code></a> (X[, allow_nan])	Throw a ValueError if X contains NaN or infinity. Бросьте ValueError, если X содержит NaN или бесконечность.
<a href="#"><code>utils.check_X_y</code></a> (X, y[, accept_sparse, ...])	Input validation for standard estimators. Подтверждение ввода для стандартных оценщиков.
<a href="#"><code>utils.check_array</code></a> (array[, accept_sparse, ...])	Input validation on an array, list, sparse matrix or similar. Проверка правильности ввода в массиве, списке, разреженной матрице и т. П.
<a href="#"><code>utils.check_scalar</code></a> (x, name, target_type[, ...])	Validate scalar parameters type and value. Проверьте скалярные параметры типа и значения.
<a href="#"><code>utils.check_consistent_length</code></a> (*arrays)	Check that all arrays have consistent first dimensions. Убедитесь, что все массивы соответствуют первым измерениям.
<a href="#"><code>utils.check_random_state</code></a> (seed)	Turn seed into a np.random.RandomState instance Превратите seed в экземпляр np.random.RandomState
<a href="#"><code>utils.class_weight.compute_class_weight</code></a> (...)	Estimate class weights for unbalanced datasets. Оцените вес класса для несбалансированных наборов данных.
<a href="#"><code>utils.class_weight.compute_sample_weight</code></a> (...)	Estimate sample weights by class for unbalanced datasets. Оцените вес выборки по классам для несбалансированных наборов данных.
<a href="#"><code>utils.deprecated</code></a> ([extra])	Decorator to mark a function or class as deprecated. Декоратор, чтобы пометить функцию или класс как устаревший.

<a href="#"><code>utils.estimator_checks.check_estimator</code></a> (Estimator)	Check if estimator adheres to scikit-learn conventions. Проверьте, придерживается ли оценщик правил scikit-learn.
<a href="#"><code>utils.extmath.safe_sparse_dot</code></a> (a, b[, ...])	Dot product that handle the sparse matrix case correctly Точечный продукт, который правильно обрабатывает случай разреженной матрицы
<a href="#"><code>utils.extmath.randomized_range_finder</code></a> (A, ...)	Computes an orthonormal matrix whose range approximates the range of A. Вычисляет ортонормированную матрицу, диапазон которой аппроксимирует диапазон A.
<a href="#"><code>utils.extmath.randomized_svd</code></a> (M, n_components)	Computes a truncated randomized SVD Вычисляет усеченное рандомизированное SVD
<a href="#"><code>utils.extmath.fast_logdet</code></a> (A)	Compute log(det(A)) for A symmetric Вычислить log (det (A)) для симметричного A
<a href="#"><code>utils.extmath.density</code></a> (w, <code>**kwargs</code> )	Compute density of a sparse vector Вычислить плотность разреженного вектора
<a href="#"><code>utils.extmath.weighted_mode</code></a> (a, w[, axis])	Returns an array of the weighted modal (most common) value in a Возвращает массив взвешенного модального (наиболее распространенного) значения в a
<a href="#"><code>utils.gen_even_slices</code></a> (n, n_packs[, n_samples])	Generator to create n_packs slices going up to n. Генератор для создания срезов n_packs до n.
<a href="#"><code>utils.graph.single_source_shortest_path_length</code></a> (...)	Return the shortest path length from source to all reachable nodes. Вернуть кратчайшую длину пути от источника ко всем достижимым узлам.
<a href="#"><code>utils.graph.shortest_path.graph_shortest_path</code></a> ()	Perform a shortest-path graph search on a positive directed or undirected graph. Выполните поиск графа по кратчайшему пути на положительно ориентированном или неориентированном графе.
<a href="#"><code>utils.indexable</code></a> ( <code>*iterables</code> )	Make arrays indexable for cross-validation. Сделайте массивы индексруемыми для перекрестной проверки.
<a href="#"><code>utils.metaestimators.if_delegate_has_method</code></a> (...)	Create a decorator for methods that are delegated to a sub-estimator Создайте декоратор для методов, которые делегированы подоценщику
<a href="#"><code>utils.multiclass.type_of_target</code></a> (y)	Determine the type of data indicated by the target. Определите тип данных, обозначенных целью.
<a href="#"><code>utils.multiclass.is_multilabel</code></a> (y)	Check if y is in a multilabel format. Проверьте, если y в многослойном формате.
<a href="#"><code>utils.multiclass.unique_labels</code></a> ( <code>*ys</code> )	Extract an ordered array of unique labels Извлечь упорядоченный массив уникальных меток
<a href="#"><code>utils.murmurhash3_32</code></a> ()	Compute the 32bit murmurhash3 of key at seed. Вычислить 32-битный ропот 3 ключа на начальном этапе.
<a href="#"><code>utils.resample</code></a> ( <code>*arrays</code> , <code>**options</code> )	Resample arrays or sparse matrices in a consistent way Повторно выбирайте массивы или разреженные матрицы
<a href="#"><code>utils.safe_indexing</code></a> (X, indices)	Return items or rows from X using indices. Вернуть элементы или строки из X, используя индексы.
<a href="#"><code>utils.safe_mask</code></a> (X, mask)	Return a mask which is safe to use on X. Верните маску, которая безопасна для использования

	на X.
<a href="#"><code>utils.safe_sqr(X[, copy])</code></a>	Element wise squaring of array-likes and sparse matrices. Элементарное возведение в квадрат массивов-подобий и разреженных матриц
<a href="#"><code>utils.shuffle(*arrays, **options)</code></a>	Shuffle arrays or sparse matrices in a consistent way Перемешивать массивы или разреженные матрицы согласованно
<a href="#"><code>utils.sparsefuncs.incr_mean_variance_axis(X, ...)</code></a>	Compute incremental mean and variance along an axis on a CSR or CSC matrix. Вычислить инкрементное среднее значение и дисперсию вдоль оси в матрице CSR или CSC.
<a href="#"><code>utils.sparsefuncs.inplace_column_scale(X, scale)</code></a>	Inplace column scaling of a CSC/CSR matrix. Масштабирование столбцов на месте матрицы CSC / CSR.
<a href="#"><code>utils.sparsefuncs.inplace_row_scale(X, scale)</code></a>	Inplace row scaling of a CSR or CSC matrix. Масштабирование строк в матрице CSR или CSC.
<a href="#"><code>utils.sparsefuncs.inplace_swap_row(X, m, n)</code></a>	Swaps two rows of a CSC/CSR matrix in-place. Меняет местами две строки матрицы CSC / CSR.
<a href="#"><code>utils.sparsefuncs.inplace_swap_column(X, m, n)</code></a>	Swaps two columns of a CSC/CSR matrix in-place. Меняет местами два столбца матрицы CSC / CSR.
<a href="#"><code>utils.sparsefuncs.mean_variance_axis(X, axis)</code></a>	Compute mean and variance along an axis on a CSR or CSC matrix Вычислить среднее значение и дисперсию вдоль оси на матрице CSR или CSC
<a href="#"><code>utils.sparsefuncs.inplace_csr_column_scale(X, ...)</code></a>	Inplace column scaling of a CSR matrix. Масштабирование столбцов матрицы CSR.
<a href="#"><code>utils.sparsefuncs.fast.inplace_csr_row_normalize_l1()</code></a>	Inplace row normalize using the l1 norm Нормализация строки на месте с использованием нормы l1
<a href="#"><code>utils.sparsefuncs.fast.inplace_csr_row_normalize_l2()</code></a>	Inplace row normalize using the l2 norm Нормализация строки на месте с использованием нормы l2
<a href="#"><code>utils.random.sample_without_replacement()</code></a>	Sample integers without replacement. Образцы целых чисел без замены.
<a href="#"><code>utils.validation.check_is_fitted(estimator, ...)</code></a>	Perform is_fitted validation for estimator. Выполните проверку is_fitted для оценщика.
<a href="#"><code>utils.validation.check_memory(memory)</code></a>	Check that memory is joblib.Memory-like. Проверьте, что память является joblib.Memory-like.
<a href="#"><code>utils.validation.check_symmetric(array[, ...])</code></a>	Make sure that array is 2D, square and symmetric. Убедитесь, что массив является 2D, квадратным и симметричным.
<a href="#"><code>utils.validation.column_or_1d(y[, warn])</code></a>	Ravel column or 1d numpy array, else raises an error Столбец Равеля или массив 1d numpy, иначе возникает ошибка
<a href="#"><code>utils.validation.has_fit_parameter(...)</code></a>	Checks whether the estimator's fit method supports the given parameter. Проверяет, поддерживает ли метод подбора оценщика данный параметр.
<a href="#"><code>utils.testing.assert_in(member, container[, msg])</code></a>	Just like self.assertTrue(a in b), but with a nicer default message. Точно так же, как self.assertTrue (a в b), но с более хорошим сообщением по умолчанию.

<a href="#"><code>utils.testing.assert_not_in</code></a> (member, container)	Just like <code>self.assertTrue(a not in b)</code> , but with a nicer default message. Точно так же, как <code>self.assertTrue (a не в b)</code> , но с более приятным сообщением по умолчанию.
<a href="#"><code>utils.testing.assert_raise_message</code></a> (...)	Helper function to test the message raised in an exception. Вспомогательная функция для проверки сообщения, возникшего в исключении.
<a href="#"><code>utils.testing.all_estimators</code></a> (...)	Get a list of all estimators from sklearn. Получить список всех оценок от sklearn.

Utilities from joblib:

<a href="#"><code>utils.parallel_backend</code></a> (backend[, n_jobs])	Change the default backend used by Parallel inside a with block. Измените бэкэнд по умолчанию, используемый Parallel внутри блока with.
<a href="#"><code>utils.register_parallel_backend</code></a> (name, factory)	Register a new Parallel backend factory. Зарегистрируйте новый параллельный бэкэнд-завод.

## Recently deprecated

## Недавно устарел

### To be removed in 0.23

### Быть удаленным в 0.23

<a href="#"><code>utils.Memory</code></a> (*args, **kwargs) <b>Attributes:</b>	
<a href="#"><code>utils.Parallel</code></a> (*args, **kwargs)	Methods
<a href="#"><code>utils.cpu_count</code></a> ()	DEPRECATED: deprecated in version 0.20.1 to be removed in version 0.23. УСТАРЕЛО: устарело в версии 0.20.1 для удаления в версии 0.23.
<a href="#"><code>utils.delayed</code></a> (function[, check_pickle])	DEPRECATED: deprecated in version 0.20.1 to be removed in version 0.23.
<a href="#"><code>metrics.calinski_harabaz_score</code></a> (X, labels)	DEPRECATED: Function 'calinski_harabaz_score' has been renamed to 'calinski_harabasz_score' and will be removed in version 0.23. УСТАРЕЛО: функция «calinski_harabaz_score» была переименована в «calinski_harabasz_score» и будет удалена в версии 0.23.
<a href="#"><code>metrics.jaccard_similarity_score</code></a> (y_true, y_pred)	Jaccard similarity coefficient score Коэффициент коэффициента подобия Жакара
<a href="#"><code>linear_model.logistic_regression_path</code></a> (X, y)	DEPRECATED: logistic_regression_path was deprecated in version 0.21 and will be removed in version 0.23.0



## To be removed in 0.22 Быть удаленным в 0.22

<a href="#"><code>covariance.GraphLasso</code></a> (*args, **kwargs)	Sparse inverse covariance estimation with an l1-penalized estimator. Оценка разреженной обратной ковариации с помощью l1-штрафной оценки.
<a href="#"><code>covariance.GraphLassoCV</code></a> (*args, **kwargs)	Sparse inverse covariance w/ cross-validated choice of the l1 penalty. Разреженная обратная ковариация с перекрестным подтверждением выбора штрафа l1.
<a href="#"><code>preprocessing.Imputer</code></a> (*args, **kwargs)	Imputation transformer for completing missing values. Преобразователь импутации для заполнения пропущенных значений.
<a href="#"><code>utils.testing.mock_mldata_url_open</code></a> (*args, ...)	Object that mocks the url_open function to fake requests to mldata. Объект, который высмеивает функцию url_open для фальсификации запросов к mldata.
<a href="#"><code>Covariance.graph_lasso</code></a> (emp_cov, alpha[, ...])	DEPRECATED: The 'graph_lasso' was renamed to 'graphical_lasso' in version 0.20 and will be removed in 0.22. УСТАРЕЛО: «graph_lasso» было переименовано в «graphical_lasso» в версии 0.20 и будет удалено в 0.22.
<a href="#"><code>datasets.fetch_mldata</code></a> (dataname[, ...])	DEPRECATED: fetch_mldata was deprecated in version 0.20 and will be removed in version 0.22. УСТАРЕЛО: fetch_mldata устарела в версии 0.20 и будет удалена в версии 0.22.
<a href="#"><code>datasets.mldata_filename</code></a> (dataname)	DEPRECATED: mldata_filename was deprecated in version 0.20 and will be removed in version 0.22.