

#_ important **Scikit-learn** Operations [+100]

General Operations:

- `sklearn.datasets.load_iris()`: Load the iris dataset.
- `sklearn.datasets.load_digits()`: Load the hand-written digits dataset.
- `sklearn.model_selection.train_test_split()`: Split datasets into training and testing subsets.

Preprocessing:

- `sklearn.preprocessing.StandardScaler()`: Standardize features by removing the mean and scaling to unit variance.
- `sklearn.preprocessing.MinMaxScaler()`: Transform features by scaling them to a given range.
- `sklearn.preprocessing.LabelEncoder()`: Encode labels with value between 0 and n_classes-1.
- `sklearn.preprocessing.OneHotEncoder()`: Convert categorical variable(s) into dummy/indicator variables.

Supervised Learning Algorithms:

Linear Models:

- `sklearn.linear_model.LinearRegression()`: Ordinary least squares linear regression.
- `sklearn.linear_model.LogisticRegression()`: Logistic regression (classification).
- `sklearn.linear_model.Ridge()`: Linear least squares with l2 regularization.

Support Vector Machines (SVM):

- `sklearn.svm.SVC()`: C-Support Vector Classification.
- `sklearn.svm.SVR()`: Epsilon-Support Vector Regression.

Nearest Neighbors:

- `sklearn.neighbors.KNeighborsClassifier()`: Classifier implementing the k-nearest neighbors vote.
- `sklearn.neighbors.KNeighborsRegressor()`: Regression based on k-nearest neighbors.

Gaussian Processes:

- `sklearn.gaussian_process.GaussianProcessRegressor()`: Gaussian process regression (GPR).
- `sklearn.gaussian_process.GaussianProcessClassifier()`: Gaussian process classification (GPC).

Decision Trees:

- `sklearn.tree.DecisionTreeClassifier()`: Decision tree classifier.
- `sklearn.tree.DecisionTreeRegressor()`: Decision tree regressor.

Ensemble Methods:

- `sklearn.ensemble.RandomForestClassifier()`: Random forest classifier.
- `sklearn.ensemble.RandomForestRegressor()`: Random forest regressor.
- `sklearn.ensemble.GradientBoostingClassifier()`: Gradient boosting classifier.
- `sklearn.ensemble.GradientBoostingRegressor()`: Gradient boosting regressor.

Neural Network Models:

- `sklearn.neural_network.MLPClassifier()`: Multi-layer perceptron classifier.
- `sklearn.neural_network.MLPRegressor()`: Multi-layer perceptron regressor.

Unsupervised Learning Algorithms:

Clustering:

- `sklearn.cluster.KMeans()`: K-Means clustering.

- `sklearn.cluster.DBSCAN()`: Density-based spatial clustering of applications with noise.
- `sklearn.cluster.AgglomerativeClustering()`: Agglomerative clustering.

Dimensionality Reduction:

- `sklearn.decomposition.PCA()`: Principal component analysis.
- `sklearn.decomposition.NMF()`: Non-negative matrix factorization.
- `sklearn.manifold.TSNE()`: t-distributed Stochastic Neighbor Embedding.

Model Selection and Evaluation:

- `sklearn.model_selection.cross_val_score()`: Evaluate a score by cross-validation.
- `sklearn.model_selection.GridSearchCV()`: Exhaustive search over specified parameter values for an estimator.
- `sklearn.model_selection.RandomizedSearchCV()`: Randomized search on hyperparameters.
- `sklearn.metrics.accuracy_score()`: Accuracy classification score.
- `sklearn.metrics.mean_squared_error()`: Mean squared error regression loss.
- `sklearn.metrics.confusion_matrix()`: Compute confusion matrix to evaluate the accuracy of a classification.
- `sklearn.metrics.roc_curve()`: Compute Receiver operating characteristic (ROC).
- `sklearn.metrics.auc()`: Compute Area Under the Curve (AUC) from prediction scores.

Pipeline:

- `sklearn.pipeline.Pipeline()`: Pipeline of transforms and a final estimator.
- `sklearn.pipeline.make_pipeline()`: Construct a Pipeline from the given estimators.

Feature Extraction:

- `sklearn.feature_extraction.text.CountVectorizer()`: Convert a collection of text documents to a matrix of token counts.
- `sklearn.feature_extraction.text.TfidfVectorizer()`: Convert a collection of raw documents to a matrix of TF-IDF features.

Feature Selection:

- `sklearn.feature_selection.SelectKBest()`: Select features according to the k highest scores.
- `sklearn.feature_selection.RFE()`: Feature ranking with recursive feature elimination.

Imbalanced Datasets:

- `sklearn.utils.class_weight.compute_class_weight()`: Estimate class weights for unbalanced datasets.

Decomposition:

- `sklearn.decomposition.TruncatedSVD()`: Dimensionality reduction using truncated SVD (aka LSA).
- `sklearn.decomposition.FastICA()`: Fast algorithm for Independent Component Analysis.

Manifold Learning:

- `sklearn.manifold.Isomap()`: Isomap embedding.
- `sklearn.manifold.MDS()`: Multi-dimensional scaling.

Dataset Transformations:

- `sklearn.preprocessing.PolynomialFeatures()`: Generate polynomial and interaction features.
- `sklearn.preprocessing.Binarizer()`: Binarize data (set feature values to 0 or 1) according to a threshold.

Validation:

- `sklearn.model_selection.StratifiedKFold()`: Stratified K-Folds cross-validator.
- `sklearn.model_selection.LeaveOneOut()`: Leave-One-Out cross-validator.

Calibration:

- `sklearn.calibration.CalibratedClassifierCV()`: Probability calibration with isotonic regression or logistic regression.

Semi-Supervised Learning:

- `sklearn.semi_supervised.LabelPropagation()`: Label Propagation classifier.
- `sklearn.semi_supervised.LabelSpreading()`: Label Spreading classifier.

Kernel Ridge Regression:

- `sklearn.kernel_ridge.KernelRidge()`: Kernel ridge regression.

Pairwise Metrics:

- `sklearn.metrics.pairwise.cosine_similarity()`: Compute cosine similarity between samples in X and Y.

Discriminant Analysis:

- `sklearn.discriminant_analysis.LinearDiscriminantAnalysis()`: Linear Discriminant Analysis.
- `sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis()`: Quadratic Discriminant Analysis.

Isolation Forest:

- `sklearn.ensemble.IsolationForest()`: Isolation Forest Algorithm.

Naive Bayes:

- `sklearn.naive_bayes.GaussianNB()`: Gaussian Naive Bayes.
- `sklearn.naive_bayes.MultinomialNB()`: Multinomial Naive Bayes.

Cross Decomposition:

- `sklearn.cross_decomposition.PLSRegression()`: PLS regression.

Nearest Centroid Classifier:

- `sklearn.neighbors.NearestCentroid()`: Nearest centroid classifier.

Neural network utilities:

- `sklearn.neural_network.BernoulliRBM()`: Bernoulli Restricted Boltzmann Machine.

Stochastic Gradient Descent:

- `sklearn.linear_model.SGDClassifier()`: Linear classifiers with SGD training.
- `sklearn.linear_model.SGDRegressor()`: Linear model fitted by minimizing a regularized empirical loss with SGD.

Multi-class and multi-label algorithms:

- `sklearn.multiclass.OneVsRestClassifier()`: One-vs-the-rest (OvR) multiclass/multilabel strategy.

Multioutput regression:

- `sklearn.multioutput.MultiOutputRegressor()`: Multioutput regression.

Multiclass-multioutput algorithms:

- `sklearn.multioutput.ClassifierChain()`: Classifier Chain.

Sparse coding:

- `sklearn.decomposition.SparseCoder()`: Sparse coding.

Covariance estimators:

- `sklearn.covariance.EmpiricalCovariance()`: Maximum likelihood covariance estimator.

Gaussian Mixture Models:

- `sklearn.mixture.GaussianMixture()`: Gaussian Mixture.

Model Evaluation & Selection:

- `sklearn.model_selection.permutation_test_score()`: Permutation test for score.

Cluster Biclustering:

- `sklearn.cluster.bicluster.SpectralBiclustering()`: Spectral Biclustering.

Sparse PCA:

- `sklearn.decomposition.SparsePCA()`: Sparse Principal Components Analysis (SparsePCA).

Voting regressor:

- `sklearn.ensemble.VotingRegressor()`: Voting regressor.

Bagging regressor:

- `sklearn.ensemble.BaggingRegressor()`: Bagging regressor.

Impute:

- `sklearn.impute.SimpleImputer()`: Basic imputation transformer.

Checking:

- `sklearn.utils.check_X_y()`: Ensure X and y have compatible shapes.

Checking Estimators:

- `sklearn.utils.estimator_checks.check_estimator()`: Check if estimator adheres to scikit-learn conventions.

Multilabel Binarizer:

- `sklearn.preprocessing.MultiLabelBinarizer()`: Transform between iterable of iterables and a multilabel format.

Cross Decomposition:

- `sklearn.cross_decomposition.CCA()`: Canonical Correlation Analysis.

Loading datasets:

- `sklearn.datasets.load_breast_cancer()`: Load breast cancer dataset.
- `sklearn.datasets.load_diabetes()`: Load diabetes dataset.
- `sklearn.datasets.load_linnerud()`: Load Linnerud dataset.

Binarize labels:

- `sklearn.preprocessing.label_binarize()`: Binarize labels in a one-vs-all fashion.

Metrics:

- `sklearn.metrics.log_loss()`: Logarithmic loss.
- `sklearn.metrics.mean_absolute_error()`: Mean absolute error regression loss.
- `sklearn.metrics.mean_squared_log_error()`: Mean squared logarithmic error regression loss.

Partial dependence plots:

- `sklearn.inspection.plot_partial_dependence()`: Partial dependence plots.

Unsupervised Neural Network:

- `sklearn.neural_network.BernoulliRBM()`: Bernoulli Restricted Boltzmann Machine.

Load sample images:

- `sklearn.datasets.load_sample_images()`: Load sample images for image manipulation.

Metrics:

- `sklearn.metrics.precision_recall_curve()`: Compute precision-recall pairs for different probability thresholds.
- `sklearn.metrics.average_precision_score()`: Compute average precision (AP) from prediction scores.

Checking:

- `sklearn.utils.check_random_state()`: Turn random state into a numpy random number generator.

Output Code:

- `sklearn.utils.murmurhash3_32()`: Hash a Python object into a 32-bit integer.

Metrics:

- `sklearn.metrics.classification_report()`: Build a text report showing the main classification metrics.
- `sklearn.metrics.cohen_kappa_score()`: Cohen's kappa: a statistic that measures inter-annotator agreement.

- `sklearn.metrics.confusion_matrix()`: Compute confusion matrix to evaluate the accuracy of a classification.
- `sklearn.metrics.hinge_loss()`: Compute (average) hinge loss.
- `sklearn.metrics.matthews_corrcoef()`: Compute the Matthews correlation coefficient (MCC) for binary classes.