**Sklearn 模块的构成**

## sklearn.base: Base classes and utility functions（基类和效用函数）

它是所有估计量的基类。

基础类：

|  |  |
| --- | --- |
| [base.BaseEstimator](http://scikit-learn.org/stable/modules/generated/sklearn.base.BaseEstimator.html" \l "sklearn.base.BaseEstimator" \o "sklearn.base.BaseEstimator" \t "http://blog.csdn.net/u010859707/article/details/_blank) | sklearn中所有估计的基础类 |
| [base.ClassifierMixin](http://scikit-learn.org/stable/modules/generated/sklearn.base.ClassifierMixin.html" \l "sklearn.base.ClassifierMixin" \o "sklearn.base.ClassifierMixin" \t "http://blog.csdn.net/u010859707/article/details/_blank) | sklearn中所有分类器的混合类 |
| [base.ClusterMixin](http://scikit-learn.org/stable/modules/generated/sklearn.base.ClusterMixin.html" \l "sklearn.base.ClusterMixin" \o "sklearn.base.ClusterMixin" \t "http://blog.csdn.net/u010859707/article/details/_blank) | sklearn中所有聚类估计器的混合类 |
| [base.RegressorMixin](http://scikit-learn.org/stable/modules/generated/sklearn.base.RegressorMixin.html" \l "sklearn.base.RegressorMixin" \o "sklearn.base.RegressorMixin" \t "http://blog.csdn.net/u010859707/article/details/_blank) | sklearn中所有回归估计器的混合类 |
| [base.TransformerMixin](http://scikit-learn.org/stable/modules/generated/sklearn.base.TransformerMixin.html" \l "sklearn.base.TransformerMixin" \o "sklearn.base.TransformerMixin" \t "http://blog.csdn.net/u010859707/article/details/_blank) | sklearn中所有变压器的混合类 |

函数：

|  |  |
| --- | --- |
| [base.clone](http://scikit-learn.org/stable/modules/generated/sklearn.base.clone.html" \l "sklearn.base.clone" \o "sklearn.base.clone" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator[, safe]) | 构造具有相同参数的新估计器 |
| config\_context(\*\*new\_config) | 为整体sklearn配置进行环境管理 |
| get\_config | 检索由set\_config确定的当前配置的值 |
| set\_config | 设置整体sklearn配置 |

## sklearn.calibration: Probability Calibration（概率校准）

它的作用是校准并预测概率。

类：

|  |  |
| --- | --- |
| [calibration.CalibratedClassifierCV](http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html" \l "sklearn.calibration.CalibratedClassifierCV" \o "sklearn.calibration.CalibratedClassifierCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 等渗回归或乙状结构的概率校准 |

sklearn.calibration.CalibratedClassifierCV的基础估计用于拟合交叉验证发生的序列集，它的测试装置用于校准，并且对每个褶皱的概率进行平均预测。

函数：

|  |  |
| --- | --- |
| calibration.calibration\_curve(y\_true,y\_prob) | 计算校准曲线的真实和预测概率 |

## sklearn.cluster: Clustering（聚类）

这个模块的作用是收集流行的无监督聚类算法。

类：

|  |  |
| --- | --- |
| [cluster.AffinityPropagation](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html" \l "sklearn.cluster.AffinityPropagation" \o "sklearn.cluster.AffinityPropagation" \t "http://blog.csdn.net/u010859707/article/details/_blank)([damping, ...]) | 执行数据的AP聚类算法 |
| [cluster.AgglomerativeClustering](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html" \l "sklearn.cluster.AgglomerativeClustering" \o "sklearn.cluster.AgglomerativeClustering" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 集聚聚类 |
| [cluster.Birch](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html" \l "sklearn.cluster.Birch" \o "sklearn.cluster.Birch" \t "http://blog.csdn.net/u010859707/article/details/_blank)([threshold, branching\_factor, ...]) | 实现Birch聚类算法 |
| [cluster.DBSCAN](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html" \l "sklearn.cluster.DBSCAN" \o "sklearn.cluster.DBSCAN" \t "http://blog.csdn.net/u010859707/article/details/_blank)([eps, min\_samples, metric, ...]) | 从矢量阵列或距离矩阵执行DBSCAN聚类 |
| [cluster.FeatureAgglomeration](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.FeatureAgglomeration.html" \l "sklearn.cluster.FeatureAgglomeration" \o "sklearn.cluster.FeatureAgglomeration" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_clusters, ...]) | 聚集特征 |
| [cluster.KMeans](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html" \l "sklearn.cluster.KMeans" \o "sklearn.cluster.KMeans" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_clusters, init, n\_init, ...]) | K-means聚类 |
| [cluster.MiniBatchKMeans](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html" \l "sklearn.cluster.MiniBatchKMeans" \o "sklearn.cluster.MiniBatchKMeans" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_clusters, init, ...]) | 小批量K均值聚类 |
| [cluster.MeanShift](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html" \l "sklearn.cluster.MeanShift" \o "sklearn.cluster.MeanShift" \t "http://blog.csdn.net/u010859707/article/details/_blank)([bandwidth, seeds, ...]) | 使用平坦内核的均值漂移聚类 |
| [cluster.SpectralClustering](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html" \l "sklearn.cluster.SpectralClustering" \o "sklearn.cluster.SpectralClustering" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_clusters, ...]) | 将聚类应用于对规范化拉普拉斯算子的投影 |

sklearn.cluster.AffinityPropagation是用于执行数据的AP(Affinity Propagation)聚类算法。

sklearn.cluster.Birch的作用是实现Birch聚类算法。这是一种高效记忆、在线内存、可以替代MiniBatchKMeans的一种算法。它构造了一个树型数据结构，其聚类中心由叶节点读出。这些可以是最终的聚类中心，或可作为输入提供给另一个如agglomerativeclustering聚类算法。

DBXCAN(Density-Based Spatial Clustering of Applications with Noise) 是一个比较有代表性的基于密度的聚类算法。与划分和层次聚类方法不同，它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类。

sklearn.cluster.FeatureAgglomeration与AgglomerativeClustering类似，只不过是递归地合并特征而不是样本。

sklearn.cluster.MeanShift中，Mean Shift聚类算法又叫均值漂移聚类，它的目的是在光滑的样本密度中发现“斑点”。这是一个基于核函数的算法，一般是指一个迭代的步骤,即先算出当前点的偏移均值,移动该点到其偏移均值,然后以此为新的起始点,继续移动,直到满足一定的条件结束。

实际运用中，谱聚类算法(SpectralClustering)在单个集群的结构高度非凸或更一般时或中心的测量或簇的分布并不是完整簇的合适描述时更加实用。

函数：

|  |  |
| --- | --- |
| [cluster.estimate\_bandwidth](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.estimate_bandwidth.html" \l "sklearn.cluster.estimate_bandwidth" \o "sklearn.cluster.estimate_bandwidth" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, quantile, ...]) | 估计带宽并用于mean-shift算法 |
| [cluster.k\_means](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.k_means.html" \l "sklearn.cluster.k_means" \o "sklearn.cluster.k_means" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, n\_clusters[, init, ...]) | K-means聚类算法 |
| [cluster.ward\_tree](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.ward_tree.html" \l "sklearn.cluster.ward_tree" \o "sklearn.cluster.ward_tree" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, connectivity, ...]) | 基于特征矩阵的区域聚类 |
| [cluster.affinity\_propagation](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.affinity_propagation.html" \l "sklearn.cluster.affinity_propagation" \o "sklearn.cluster.affinity_propagation" \t "http://blog.csdn.net/u010859707/article/details/_blank)(S[, ...]) | 执行AP聚类算法 |
| [cluster.dbscan](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.dbscan.html" \l "sklearn.cluster.dbscan" \o "sklearn.cluster.dbscan" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, eps, min\_samples, ...]) | 从矢量阵列或距离矩阵执行DBSCAN聚类 |
| [cluster.mean\_shift](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.mean_shift.html" \l "sklearn.cluster.mean_shift" \o "sklearn.cluster.mean_shift" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, bandwidth, seeds, ...]) | 使用平坦内核对数据执行均值漂移聚类算法 |
| [cluster.spectral\_clustering](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.spectral_clustering.html" \l "sklearn.cluster.spectral_clustering" \o "sklearn.cluster.spectral_clustering" \t "http://blog.csdn.net/u010859707/article/details/_blank)(affinity[, ...]) | 将聚类应用于对规范化拉普拉斯算子的投影 |

## sklearn.cluster.bicluster: Biclustering（双聚类）

光谱双聚类算法。

类：

|  |  |
| --- | --- |
| [SpectralBiclustering](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.bicluster.SpectralBiclustering.html" \l "sklearn.cluster.bicluster.SpectralBiclustering" \o "sklearn.cluster.bicluster.SpectralBiclustering" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_clusters, method, ...]) | 光谱双聚类（Kluger，2003） |
| [SpectralCoclustering](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.bicluster.SpectralCoclustering.html" \l "sklearn.cluster.bicluster.SpectralCoclustering" \o "sklearn.cluster.bicluster.SpectralCoclustering" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_clusters, ...]) | 光谱共聚焦算法（Dhillon，2001） |

sklearn.cluster.bicluster.SpectralBiclustering中，在数据具有基本棋盘结构的假设下对行和列进行分区。比如，假如有两个行分区和三个列分区，那么每行就属于三个双聚类，每列属于两个双聚类。相应的行和柱标签矢量的外积给出了这样的棋盘结构。

sklearn.cluster.bicluster.SpectralCoclustering里，聚类数组X的行和列来解决由X创建出来的偶图的松弛的归一化割问题，如下：行顶点i和列顶点j的边的权为X[i,j]。

## sklearn.covariance: Covariance Estimators（协方差估计）

该sklearn.covariance模块包括方法和算法，以大概地估计给定一组点的特征的协方差。定义为协方差的倒数的精度矩阵也被估计。协方差估计与高斯图形模型的理论密切相关。

类：

|  |  |
| --- | --- |
| [covariance.EmpiricalCovariance](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.EmpiricalCovariance.html" \l "sklearn.covariance.EmpiricalCovariance" \o "sklearn.covariance.EmpiricalCovariance" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 最大似然协方差估计 |
| [covariance.EllipticEnvelope](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.EllipticEnvelope.html" \l "sklearn.covariance.EllipticEnvelope" \o "sklearn.covariance.EllipticEnvelope" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 用于检测高斯分布数据集中异常值的对象 |
| [covariance.GraphLasso](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.GraphLasso.html" \l "sklearn.covariance.GraphLasso" \o "sklearn.covariance.GraphLasso" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, mode, tol, ...]) | 具有l1惩罚估计量的稀疏逆协方差估计 |
| [covariance.GraphLassoCV](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.GraphLassoCV.html" \l "sklearn.covariance.GraphLassoCV" \o "sklearn.covariance.GraphLassoCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alphas, ...]) | 稀疏逆协方差/交叉验证的l1罚款选择 |
| [covariance.LedoitWolf](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.LedoitWolf.html" \l "sklearn.covariance.LedoitWolf" \o "sklearn.covariance.LedoitWolf" \t "http://blog.csdn.net/u010859707/article/details/_blank)([store\_precision, ...]) | LedoitWolf估计 |
| [covariance.MinCovDet](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.MinCovDet.html" \l "sklearn.covariance.MinCovDet" \o "sklearn.covariance.MinCovDet" \t "http://blog.csdn.net/u010859707/article/details/_blank)([store\_precision, ...]) | 最小协方差决定因素（MCD）：协方差的robust估计 |
| [covariance.OAS](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.OAS.html" \l "sklearn.covariance.OAS" \o "sklearn.covariance.OAS" \t "http://blog.csdn.net/u010859707/article/details/_blank)([store\_precision, ...]) | Oracle近似收缩估计 |
| [covariance.ShrunkCovariance](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.ShrunkCovariance.html" \l "sklearn.covariance.ShrunkCovariance" \o "sklearn.covariance.ShrunkCovariance" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 协变量估计与收缩 |

sklearn.covariance.LedoitWolf中，Ledoit-Wolf是一种特别的收缩形式，其中收缩系数用O来计算。

sklearn.covariance.MinCovDet中，MCD协方差估计应用于高斯分布数据，但仍然可能与单峰对称分布得出的数据有关。它并不能与多模态数据一起使用（这种算法适用于MCD对象但在这种例子中很可能会失败）。所以，在处理多模态数据集时应考虑投影寻踪方法。

函数：

|  |  |
| --- | --- |
| [covariance.empirical\_covariance](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.empirical_covariance.html" \l "sklearn.covariance.empirical_covariance" \o "sklearn.covariance.empirical_covariance" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, ...]) | 计算最大似然协方差估计 |
| [covariance.ledoit\_wolf](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.ledoit_wolf.html" \l "sklearn.covariance.ledoit_wolf" \o "sklearn.covariance.ledoit_wolf" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, assume\_centered, ...]) | 估计缩小的Ledoit-Wolf协方差矩阵 |
| [covariance.shrunk\_covariance](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.shrunk_covariance.html" \l "sklearn.covariance.shrunk_covariance" \o "sklearn.covariance.shrunk_covariance" \t "http://blog.csdn.net/u010859707/article/details/_blank)(emp\_cov[, ...]) | 计算对角线上收缩的协方差矩阵 |
| [covariance.oas](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.oas.html" \l "sklearn.covariance.oas" \o "sklearn.covariance.oas" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, assume\_centered]) | 使用Oracle近似收缩算法估计协方差 |
| [covariance.graph\_lasso](http://scikit-learn.org/stable/modules/generated/sklearn.covariance.graph_lasso.html" \l "sklearn.covariance.graph_lasso" \o "sklearn.covariance.graph_lasso" \t "http://blog.csdn.net/u010859707/article/details/_blank)(emp\_cov, alpha[, ...]) | l1惩罚协方差估计 |

## sklearn.cross\_decomposition: Cross decomposition（交叉分解）

交叉分解模块包含两大类算法：偏最小二乘法（PLS）和典型相关分析（CCA）。这些算法可用于发现两个多元数据集之间的线性关系：拟合方法的x和y参数是二维数组。

类：

|  |  |
| --- | --- |
| [cross\_decomposition.PLSRegression](http://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSRegression.html" \l "sklearn.cross_decomposition.PLSRegression" \o "sklearn.cross_decomposition.PLSRegression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | PLS回归 |
| [cross\_decomposition.PLSCanonical](http://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSCanonical.html" \l "sklearn.cross_decomposition.PLSCanonical" \o "sklearn.cross_decomposition.PLSCanonical" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | PLSCanonical实现了原始Wold算法的2块规范PLS [Tenenhaus 1998] p.204，在[Wegelin 2000]中被称为PLS-C2A |
| [cross\_decomposition.CCA](http://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.CCA.html" \l "sklearn.cross_decomposition.CCA" \o "sklearn.cross_decomposition.CCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | CCA规范相关分析 |
| [cross\_decomposition.PLSSVD](http://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSSVD.html" \l "sklearn.cross_decomposition.PLSSVD" \o "sklearn.cross_decomposition.PLSSVD" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 部分最小二乘SVD |

## sklearn.datasets: Datasets（数据集）

该sklearn.datasets模块包括用于加载数据集的实用程序，包括加载和获取流行参考数据集的方法。它还具有一些人工数据生成器。

装载机：

|  |  |
| --- | --- |
| [datasets.clear\_data\_home](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.clear_data_home.html" \l "sklearn.datasets.clear_data_home" \o "sklearn.datasets.clear_data_home" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home]) | 删除数据家庭缓存的所有内容 |
| [datasets.get\_data\_home](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.get_data_home.html" \l "sklearn.datasets.get_data_home" \o "sklearn.datasets.get_data_home" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home]) | 返回scikit-learn数据目录的路径 |
| [datasets.fetch\_20newsgroups](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html" \l "sklearn.datasets.fetch_20newsgroups" \o "sklearn.datasets.fetch_20newsgroups" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home, ...]) | 加载20个新闻组数据集中的文件名和数据 |
| [datasets.fetch\_20newsgroups\_vectorized](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups_vectorized.html" \l "sklearn.datasets.fetch_20newsgroups_vectorized" \o "sklearn.datasets.fetch_20newsgroups_vectorized" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 加载20个新闻组数据集并将其转换为tf-idf向量 |
| [datasets.load\_boston](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html" \l "sklearn.datasets.load_boston" \o "sklearn.datasets.load_boston" \t "http://blog.csdn.net/u010859707/article/details/_blank)([return\_X\_y]) | 加载并返回波士顿房价数据集（回归） |
| [datasets.load\_breast\_cancer](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html" \l "sklearn.datasets.load_breast_cancer" \o "sklearn.datasets.load_breast_cancer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([return\_X\_y]) | 加载并返回乳腺癌威斯康星数据集（分类） |
| [datasets.load\_diabetes](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html" \l "sklearn.datasets.load_diabetes" \o "sklearn.datasets.load_diabetes" \t "http://blog.csdn.net/u010859707/article/details/_blank)([return\_X\_y]) | 加载并返回糖尿病数据集（回归） |
| [datasets.load\_digits](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html" \l "sklearn.datasets.load_digits" \o "sklearn.datasets.load_digits" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_class, return\_X\_y]) | 加载并返回数字数据集（分类） |
| [datasets.load\_files](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_files.html" \l "sklearn.datasets.load_files" \o "sklearn.datasets.load_files" \t "http://blog.csdn.net/u010859707/article/details/_blank)(container\_path[, ...]) | 加载文本文件类别的子文件夹名称 |
| [datasets.load\_iris](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html" \l "sklearn.datasets.load_iris" \o "sklearn.datasets.load_iris" \t "http://blog.csdn.net/u010859707/article/details/_blank)([return\_X\_y]) | 加载并返回虹膜数据集（分类） |
| [datasets.fetch\_lfw\_pairs](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_lfw_pairs.html" \l "sklearn.datasets.fetch_lfw_pairs" \o "sklearn.datasets.fetch_lfw_pairs" \t "http://blog.csdn.net/u010859707/article/details/_blank)([subset, ...]) | 人脸识别库(LFW)中一对的装载机 |
| [datasets.fetch\_lfw\_people](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_lfw_people.html" \l "sklearn.datasets.fetch_lfw_people" \o "sklearn.datasets.fetch_lfw_people" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home, ...]) | 人脸识别库(LFW)中人的装载机 |
| [datasets.load\_linnerud](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_linnerud.html" \l "sklearn.datasets.load_linnerud" \o "sklearn.datasets.load_linnerud" \t "http://blog.csdn.net/u010859707/article/details/_blank)([return\_X\_y]) | 加载并返回linnerud数据集（多元回归） |
| [datasets.mldata\_filename](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.mldata_filename.html" \l "sklearn.datasets.mldata_filename" \o "sklearn.datasets.mldata_filename" \t "http://blog.csdn.net/u010859707/article/details/_blank)(dataname) | 转换[mldata.org](http://mldata.org/" \t "http://blog.csdn.net/u010859707/article/details/_blank)文件名中的数据集的原始名称 |
| [datasets.fetch\_mldata](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_mldata.html" \l "sklearn.datasets.fetch_mldata" \o "sklearn.datasets.fetch_mldata" \t "http://blog.csdn.net/u010859707/article/details/_blank)(dataname[, ...]) | 获取[mldata.org](http://mldata.org/" \t "http://blog.csdn.net/u010859707/article/details/_blank)数据集 |
| [datasets.fetch\_olivetti\_faces](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html" \l "sklearn.datasets.fetch_olivetti_faces" \o "sklearn.datasets.fetch_olivetti_faces" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home, ...]) | 来自AT＆T的Olivetti人脸数据集的装载机 |
| [datasets.fetch\_california\_housing](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html" \l "sklearn.datasets.fetch_california_housing" \o "sklearn.datasets.fetch_california_housing" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 来自StatLib的加州住房数据集的装载机 |
| [datasets.fetch\_covtype](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_covtype.html" \l "sklearn.datasets.fetch_covtype" \o "sklearn.datasets.fetch_covtype" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home, ...]) | 加载植被型数据集，必要时下载 |
| [datasets.fetch\_kddcup99](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_kddcup99.html" \l "sklearn.datasets.fetch_kddcup99" \o "sklearn.datasets.fetch_kddcup99" \t "http://blog.csdn.net/u010859707/article/details/_blank)([subset, shuffle, ...]) | 加载并返回kddcup 99数据集（分类） |
| [datasets.fetch\_rcv1](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_rcv1.html" \l "sklearn.datasets.fetch_rcv1" \o "sklearn.datasets.fetch_rcv1" \t "http://blog.csdn.net/u010859707/article/details/_blank)([data\_home, subset, ...]) | 加载RCV1多标签数据集，必要时下载 |
| [datasets.load\_mlcomp](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_mlcomp.html" \l "sklearn.datasets.load_mlcomp" \o "sklearn.datasets.load_mlcomp" \t "http://blog.csdn.net/u010859707/article/details/_blank)(name\_or\_id[, set\_, ...]) | 加载从[http://mlcomp.org](http://mlcomp.org/" \t "http://blog.csdn.net/u010859707/article/details/_blank)下载的数据集 |
| [datasets.load\_sample\_image](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_sample_image.html" \l "sklearn.datasets.load_sample_image" \o "sklearn.datasets.load_sample_image" \t "http://blog.csdn.net/u010859707/article/details/_blank)(image\_name) | 加载单个样本图像的numpy数组 |
| [datasets.load\_sample\_images](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_sample_images.html" \l "sklearn.datasets.load_sample_images" \o "sklearn.datasets.load_sample_images" \t "http://blog.csdn.net/u010859707/article/details/_blank)() | 加载样品图像进行图像处理 |
| [datasets.fetch\_species\_distributions](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_species_distributions.html" \l "sklearn.datasets.fetch_species_distributions" \o "sklearn.datasets.fetch_species_distributions" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 来自Phillips等的物种分布数据集的装载机 |
| [datasets.load\_svmlight\_file](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_svmlight_file.html" \l "sklearn.datasets.load_svmlight_file" \o "sklearn.datasets.load_svmlight_file" \t "http://blog.csdn.net/u010859707/article/details/_blank)(f[, n\_features, ...]) | 将svmlight / libsvm格式的数据集加载到稀疏CSR矩阵中 |
| [datasets.load\_svmlight\_files](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_svmlight_files.html" \l "sklearn.datasets.load_svmlight_files" \o "sklearn.datasets.load_svmlight_files" \t "http://blog.csdn.net/u010859707/article/details/_blank)(files[, ...]) | 从SVMlight格式的多个文件加载数据集 |
| [datasets.dump\_svmlight\_file](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.dump_svmlight_file.html" \l "sklearn.datasets.dump_svmlight_file" \o "sklearn.datasets.dump_svmlight_file" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y, f[, ...]) | 以svmlight / libsvm文件格式转储数据集 |
| datasets.load\_wine([return\_X\_y]) | 加载并返回酒类数据集（分类） |

sklearn.datasets.dump\_svmlight\_file和sklearn.datasets.load\_svmlight\_file里，这种格式是基于文本的格式，每行有一个示例。它不存储零值特性，因此适合于备件数据集。

sklearn.datasets.get\_data\_home中，一些大型数据集加载器使用此文件夹避免多次下载数据。

Sklearn.datasets.load\_swmlight\_files函数相当于将load\_svmlight\_file映射到一系列文件上，除了它的结果被连接成一个单一的平面列表，并且样本向量被限制为所有的都具有相同数量的特征。

样本生成器：

|  |  |
| --- | --- |
| [datasets.make\_blobs](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html" \l "sklearn.datasets.make_blobs" \o "sklearn.datasets.make_blobs" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, n\_features, ...]) | 生成用于聚类的各向同性高斯斑点 |
| [datasets.make\_classification](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html" \l "sklearn.datasets.make_classification" \o "sklearn.datasets.make_classification" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, ...]) | 生成随机n类分类问题 |
| [datasets.make\_circles](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html" \l "sklearn.datasets.make_circles" \o "sklearn.datasets.make_circles" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, shuffle, ...]) | 用二维结构制作一个包含较小圆的大圆 |
| [datasets.make\_friedman1](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_friedman1.html" \l "sklearn.datasets.make_friedman1" \o "sklearn.datasets.make_friedman1" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, ...]) | 生成“Friedman＃1”回归问题 |
| [datasets.make\_friedman2](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_friedman2.html" \l "sklearn.datasets.make_friedman2" \o "sklearn.datasets.make_friedman2" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, noise, ...]) | 生成“Friedman＃2”回归问题 |
| [datasets.make\_friedman3](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_friedman3.html" \l "sklearn.datasets.make_friedman3" \o "sklearn.datasets.make_friedman3" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, noise, ...]) | 生成“Friedman＃3”回归问题 |
| [datasets.make\_gaussian\_quantiles](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_gaussian_quantiles.html" \l "sklearn.datasets.make_gaussian_quantiles" \o "sklearn.datasets.make_gaussian_quantiles" \t "http://blog.csdn.net/u010859707/article/details/_blank)([mean, ...]) | 通过分位数生成各向同性高斯和标签样本 |
| [datasets.make\_hastie\_10\_2](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_hastie_10_2.html" \l "sklearn.datasets.make_hastie_10_2" \o "sklearn.datasets.make_hastie_10_2" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, ...]) | 生成Hastie等人使用的二进制分类数据 |
| [datasets.make\_low\_rank\_matrix](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_low_rank_matrix.html" \l "sklearn.datasets.make_low_rank_matrix" \o "sklearn.datasets.make_low_rank_matrix" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, ...]) | 生成具有钟形奇异值的大多数低阶矩阵 |
| [datasets.make\_moons](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html" \l "sklearn.datasets.make_moons" \o "sklearn.datasets.make_moons" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, shuffle, ...]) | 使两个交错半圈 |
| [datasets.make\_multilabel\_classification](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_multilabel_classification.html" \l "sklearn.datasets.make_multilabel_classification" \o "sklearn.datasets.make_multilabel_classification" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 生成一个随机多标签分类问题 |
| [datasets.make\_regression](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html" \l "sklearn.datasets.make_regression" \o "sklearn.datasets.make_regression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, ...]) | 生成随机回归问题 |
| [datasets.make\_s\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_s_curve.html" \l "sklearn.datasets.make_s_curve" \o "sklearn.datasets.make_s_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, noise, ...]) | 生成S曲线数据集 |
| [datasets.make\_sparse\_coded\_signal](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_sparse_coded_signal.html" \l "sklearn.datasets.make_sparse_coded_signal" \o "sklearn.datasets.make_sparse_coded_signal" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n\_samples, ...) | 生成信号作为字典元素的稀疏组合 |
| [datasets.make\_sparse\_spd\_matrix](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_sparse_spd_matrix.html" \l "sklearn.datasets.make_sparse_spd_matrix" \o "sklearn.datasets.make_sparse_spd_matrix" \t "http://blog.csdn.net/u010859707/article/details/_blank)([dim, ...]) | 产生一个稀疏的对称确定正矩阵 |
| [datasets.make\_sparse\_uncorrelated](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_sparse_uncorrelated.html" \l "sklearn.datasets.make_sparse_uncorrelated" \o "sklearn.datasets.make_sparse_uncorrelated" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 生成稀疏不相关设计的随机回归问题 |
| [datasets.make\_spd\_matrix](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_spd_matrix.html" \l "sklearn.datasets.make_spd_matrix" \o "sklearn.datasets.make_spd_matrix" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n\_dim[, random\_state]) | 产生一个随机对称，正定矩阵 |
| [datasets.make\_swiss\_roll](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_swiss_roll.html" \l "sklearn.datasets.make_swiss_roll" \o "sklearn.datasets.make_swiss_roll" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_samples, noise, ...]) | 生成瑞士卷数据集 |
| [datasets.make\_biclusters](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_biclusters.html" \l "sklearn.datasets.make_biclusters" \o "sklearn.datasets.make_biclusters" \t "http://blog.csdn.net/u010859707/article/details/_blank)(shape, n\_clusters) | 生成一个具有恒定块对角线结构的阵列，用于双聚集 |
| [datasets.make\_checkerboard](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_checkerboard.html" \l "sklearn.datasets.make_checkerboard" \o "sklearn.datasets.make_checkerboard" \t "http://blog.csdn.net/u010859707/article/details/_blank)(shape, n\_clusters) | 生成具有块棋盘结构的数组，用于双聚集 |

sklearn.datasets.make\_circles能够可视化聚类和分类算法的简单玩具数据集。

## sklearn.decomposition: Matrix Decomposition（矩阵分解）

该sklearn.decomposition模块包括矩阵分解算法，其中包括PCA，NMF或ICA。该模块的大多数算法可以被认为是降维技术。

类：

|  |  |
| --- | --- |
| [decomposition.PCA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html" \l "sklearn.decomposition.PCA" \o "sklearn.decomposition.PCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, copy, ...]) | 主成分分析（PCA） |
| [decomposition.IncrementalPCA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.IncrementalPCA.html" \l "sklearn.decomposition.IncrementalPCA" \o "sklearn.decomposition.IncrementalPCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 增量主成分分析（IPCA） |
| [decomposition.KernelPCA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.KernelPCA.html" \l "sklearn.decomposition.KernelPCA" \o "sklearn.decomposition.KernelPCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 内核主成分分析（KPCA） |
| [decomposition.FactorAnalysis](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FactorAnalysis.html" \l "sklearn.decomposition.FactorAnalysis" \o "sklearn.decomposition.FactorAnalysis" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 因子分析（FA） |
| [decomposition.FastICA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html" \l "sklearn.decomposition.FastICA" \o "sklearn.decomposition.FastICA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | FastICA：独立分量分析的快速算法 |
| [decomposition.TruncatedSVD](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html" \l "sklearn.decomposition.TruncatedSVD" \o "sklearn.decomposition.TruncatedSVD" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 使用截断的SVD（也称为LSA）进行尺寸缩小 |
| [decomposition.NMF](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html" \l "sklearn.decomposition.NMF" \o "sklearn.decomposition.NMF" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, init, ...]) | 非负矩阵因子分解（NMF） |
| [decomposition.SparsePCA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparsePCA.html" \l "sklearn.decomposition.SparsePCA" \o "sklearn.decomposition.SparsePCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 稀疏主成分分析（SparsePCA） |
| [decomposition.MiniBatchSparsePCA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.MiniBatchSparsePCA.html" \l "sklearn.decomposition.MiniBatchSparsePCA" \o "sklearn.decomposition.MiniBatchSparsePCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 小批量稀疏主成分分析 |
| [decomposition.SparseCoder](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparseCoder.html" \l "sklearn.decomposition.SparseCoder" \o "sklearn.decomposition.SparseCoder" \t "http://blog.csdn.net/u010859707/article/details/_blank)(dictionary[, ...]) | 稀疏编码 |
| [decomposition.DictionaryLearning](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.DictionaryLearning.html" \l "sklearn.decomposition.DictionaryLearning" \o "sklearn.decomposition.DictionaryLearning" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 词典学习 |
| [decomposition.MiniBatchDictionaryLearning](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.MiniBatchDictionaryLearning.html" \l "sklearn.decomposition.MiniBatchDictionaryLearning" \o "sklearn.decomposition.MiniBatchDictionaryLearning" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 小批量字典学习 |
| [decomposition.LatentDirichletAllocation](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html" \l "sklearn.decomposition.LatentDirichletAllocation" \o "sklearn.decomposition.LatentDirichletAllocation" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | LDA与在线变分贝叶斯算法 |

sklearn.decomposition.DictionaryLearning找到一个字典（一组原子），最好用稀疏代码来表示数据，用于解决优化问题。

Sklearn.decomposition.FactorAnalysis构建一个具有高斯隐变量的简单线性生成模型。实验假设是由低维潜在因素和添加噪音的线性转换引起的。在不损失一般性的情况下，因子根据零均值和单位协方差的高斯分布。其噪音也是零均值并且有一个任意的对角协方差矩阵。

Sklearn.decomposition.IncrementalPCA中，IPCA使用中心数据的奇异值分解来进行线性降维，仅保留最重要的奇异向量将数据投射到低维空间。根据输入数据的大小，该算法可以比PCA节省更多的内存。

sklearn.decomposition.MiniBatchSparsePCA，找到能够优化重建数据的稀疏组件集。稀疏量是由参数给出的L1惩罚系数控制的。

sklearn.decomposition.TruncatedSVD中，该变压器通过截断奇异值分解实现线性降维(SVD)。与PCA相反，该估计器在计算奇异值分解之前不将数据居中。这意味着它可以有效地与scipy.sparse矩阵一同工作。

函数：

|  |  |
| --- | --- |
| [decomposition.fastica](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.fastica.html" \l "sklearn.decomposition.fastica" \o "sklearn.decomposition.fastica" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, n\_components, ...]) | 执行快速独立成分分析 |
| [decomposition.dict\_learning](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.dict_learning.html" \l "sklearn.decomposition.dict_learning" \o "sklearn.decomposition.dict_learning" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, n\_components, ...) | 解决词典学习矩阵分解问题 |
| [decomposition.dict\_learning\_online](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.dict_learning_online.html" \l "sklearn.decomposition.dict_learning_online" \o "sklearn.decomposition.dict_learning_online" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, ...]) | 在线解决词典学习矩阵分解问题 |
| [decomposition.sparse\_encode](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.sparse_encode.html" \l "sklearn.decomposition.sparse_encode" \o "sklearn.decomposition.sparse_encode" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, dictionary[, ...]) | 稀疏编码 |

## sklearn.discriminant\_analysis: Discriminant Analysis（判别分析）

线性判别分析和二次判别分析。

类：

|  |  |
| --- | --- |
| [discriminant\_analysis.LinearDiscriminantAnalysis](http://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html" \l "sklearn.discriminant_analysis.LinearDiscriminantAnalysis" \o "sklearn.discriminant_analysis.LinearDiscriminantAnalysis" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 线性判别分析 |
| [discriminant\_analysis.QuadraticDiscriminantAnalysis](http://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis.html" \l "sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis" \o "sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 二次判别分析 |

线性判别分析(Linear Discriminant Analysis)简称LDA，是分类算法中的一种。LDA通过对历史数据进行投影，以保证投影后同一类别的数据尽量靠近，不同类别的数据尽量分开。并生成线性判别模型对新生成的数据进行分离和预测。

与线性判别分析类似，二次判别分析是另外一种线性判别分析算法，二者拥有类似的算法特征，区别仅在于：当不同分类样本的协方差矩阵相同时，使用线性判别分析；当不同分类样本的协方差矩阵不同时，则应该使用二次判别。

## sklearn.dummy: Dummy estimators（虚拟估计）

当进行监督学习时，一个简单明智的check包括：使用不同的规则比较一个estimator。

类：

|  |  |
| --- | --- |
| [dummy.DummyClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html" \l "sklearn.dummy.DummyClassifier" \o "sklearn.dummy.DummyClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([strategy, ...]) | DummyClassifier是使用简单规则进行预测的分类器 |
| [dummy.DummyRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyRegressor.html" \l "sklearn.dummy.DummyRegressor" \o "sklearn.dummy.DummyRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([strategy, constant, ...]) | DummyRegressor是使用简单规则进行预测的倒数 |

sklearn.dummy.DummyClassifier，它的分类的作用与其它真实分类相比是作为一个简单的基线。所以不要将它用于解决真实问题。同样的，sklearn.dummy.DummyRegressor，它的回归与其他真实回归相比就是一个简单的基准，因此也不用它来解决真实问题。

## sklearn.ensemble: Ensemble Methods（集成方法）

该[sklearn.ensemble](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.ensemble" \o "sklearn.ensemble" \t "http://blog.csdn.net/u010859707/article/details/_blank)模块包括用于分类，回归和异常检测的基于集成的方法。

类：

|  |  |
| --- | --- |
| [ensemble.AdaBoostClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html" \l "sklearn.ensemble.AdaBoostClassifier" \o "sklearn.ensemble.AdaBoostClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 一个AdaBoost分类器 |
| [ensemble.AdaBoostRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html" \l "sklearn.ensemble.AdaBoostRegressor" \o "sklearn.ensemble.AdaBoostRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([base\_estimator, ...]) | AdaBoost回归器 |
| [ensemble.BaggingClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html" \l "sklearn.ensemble.BaggingClassifier" \o "sklearn.ensemble.BaggingClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([base\_estimator, ...]) | Bagging分类器 |
| [ensemble.BaggingRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html" \l "sklearn.ensemble.BaggingRegressor" \o "sklearn.ensemble.BaggingRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([base\_estimator, ...]) | Bagging回归器 |
| [ensemble.ExtraTreesClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html" \l "sklearn.ensemble.ExtraTreesClassifier" \o "sklearn.ensemble.ExtraTreesClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 一个额外的树分类器 |
| [ensemble.ExtraTreesRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html" \l "sklearn.ensemble.ExtraTreesRegressor" \o "sklearn.ensemble.ExtraTreesRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_estimators, ...]) | 一个额外的树回归器 |
| [ensemble.GradientBoostingClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html" \l "sklearn.ensemble.GradientBoostingClassifier" \o "sklearn.ensemble.GradientBoostingClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([loss, ...]) | 梯度提升分类 |
| [ensemble.GradientBoostingRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html" \l "sklearn.ensemble.GradientBoostingRegressor" \o "sklearn.ensemble.GradientBoostingRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([loss, ...]) | 渐变提升回归 |
| [ensemble.IsolationForest](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html" \l "sklearn.ensemble.IsolationForest" \o "sklearn.ensemble.IsolationForest" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_estimators, ...]) | 隔离森林算法 |
| [ensemble.RandomForestClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html" \l "sklearn.ensemble.RandomForestClassifier" \o "sklearn.ensemble.RandomForestClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 随机森林分类器 |
| [ensemble.RandomTreesEmbedding](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomTreesEmbedding.html" \l "sklearn.ensemble.RandomTreesEmbedding" \o "sklearn.ensemble.RandomTreesEmbedding" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 一个完全随机的树的集成 |
| [ensemble.RandomForestRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html" \l "sklearn.ensemble.RandomForestRegressor" \o "sklearn.ensemble.RandomForestRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 随机森林回归器 |
| [ensemble.VotingClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html" \l "sklearn.ensemble.VotingClassifier" \o "sklearn.ensemble.VotingClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimators[, ...]) | 软投票/多数规则分类器 |

Adaboost是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器（强分类器）。

bagging是一种用来提高学习算法准确度的方法，这种方法通过构造一个预测函数系列，然后以一定的方式将它们组合成一个预测函数。Bagging要求“不稳定”（不稳定是指数据集的小的变动能够使得分类结果的显著的变动）的分类方法。比如：决策树，神经网络算法。

sklearn.ensemble.IsolationForest使用IsolationForest算法返回每个样本的异常值。sklearn.ensemble.RandomTreesEmbedding进行从数据集到高维稀疏表示的非监督转换，一个数据点的编码是根据它所被分类的树的叶节点来确定的。

部分依赖：树组合的部分依赖图

|  |  |
| --- | --- |
| [ensemble.partial\_dependence.partial\_dependence](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.partial_dependence.partial_dependence.html" \l "sklearn.ensemble.partial_dependence.partial_dependence" \o "sklearn.ensemble.partial_dependence.partial_dependence" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 部分依赖target\_variables |
| [ensemble.partial\_dependence.plot\_partial\_dependence](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.partial_dependence.plot_partial_dependence.html" \l "sklearn.ensemble.partial_dependence.plot_partial_dependence" \o "sklearn.ensemble.partial_dependence.plot_partial_dependence" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 部分依赖图features |

## sklearn.exceptions: Exceptions and warnings（异常和警告）

该sklearn.exceptions模块包括在scikit学习中使用的所有自定义警告和错误类。

类：

|  |  |
| --- | --- |
| [exceptions.NotFittedError](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.NotFittedError.html" \l "sklearn.exceptions.NotFittedError" \o "sklearn.exceptions.NotFittedError" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 如果在拟合前使用估计器，则提升异常类 |
| [exceptions.ChangedBehaviorWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.ChangedBehaviorWarning.html" \l "sklearn.exceptions.ChangedBehaviorWarning" \o "sklearn.exceptions.ChangedBehaviorWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 用于通知用户任何行为变化的警告类 |
| [exceptions.ConvergenceWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.ConvergenceWarning.html" \l "sklearn.exceptions.ConvergenceWarning" \o "sklearn.exceptions.ConvergenceWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 捕捉收敛问题的自定义警告 |
| [exceptions.DataConversionWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.DataConversionWarning.html" \l "sklearn.exceptions.DataConversionWarning" \o "sklearn.exceptions.DataConversionWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 警告用于通知代码中发生的隐式数据转换 |
| [exceptions.DataDimensionalityWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.DataDimensionalityWarning.html" \l "sklearn.exceptions.DataDimensionalityWarning" \o "sklearn.exceptions.DataDimensionalityWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 自定义警告，以通知数据维度的潜在问题 |
| [exceptions.EfficiencyWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.EfficiencyWarning.html" \l "sklearn.exceptions.EfficiencyWarning" \o "sklearn.exceptions.EfficiencyWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 用于通知用户效率低下的警告 |
| [exceptions.FitFailedWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.FitFailedWarning.html" \l "sklearn.exceptions.FitFailedWarning" \o "sklearn.exceptions.FitFailedWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 如果在拟合估计器时出现错误，则使用警告类 |
| [exceptions.NonBLASDotWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.NonBLASDotWarning.html" \l "sklearn.exceptions.NonBLASDotWarning" \o "sklearn.exceptions.NonBLASDotWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 点操作不使用BLAS时使用的警告 |
| [exceptions.UndefinedMetricWarning](http://scikit-learn.org/stable/modules/generated/sklearn.exceptions.UndefinedMetricWarning.html" \l "sklearn.exceptions.UndefinedMetricWarning" \o "sklearn.exceptions.UndefinedMetricWarning" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 度量无效时使用的警告 |

sklearn.exceptions.DataConversionWarning中，当一些输入数据需要以不符合用户期望的方式转换或解释时，就会发生此警告。

sklearn.exceptions.DataDimensionalityWarning，举一个例子，在随机投影时，警告会在组件的数量量化了目标投影空间的维数时出现，意味着问题的维数不会减少。

sklearn.exceptions.EfficiencyWarning中，此警告通知用户其效率可能不是最佳的，因为有些原因可能会被视为警告消息的一部分，这可能会被继承到一个更具体的警告类。

sklearn.exceptions.NonBLASDotWarning里，这个警告用于通知用户BLAS不用于点操作，因此效率的可能会收到影响。

## sklearn.feature\_extraction: Feature Extraction（特征提取）

该sklearn.feature\_extraction模块处理原始数据的特征提取。它目前包括从文本和图像中提取特征的方法。

类：

|  |  |
| --- | --- |
| [feature\_extraction.DictVectorizer](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html" \l "sklearn.feature_extraction.DictVectorizer" \o "sklearn.feature_extraction.DictVectorizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([dtype, ...]) | 将特征值映射列表转换为向量 |
| [feature\_extraction.FeatureHasher](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.FeatureHasher.html" \l "sklearn.feature_extraction.FeatureHasher" \o "sklearn.feature_extraction.FeatureHasher" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 实现哈希功能，又称哈希技巧 |

第一个类中，转换器将字典类对象的映射列表的特征名称转化为Numpy数组或scipy.sparse矩阵的特征值，用于sklearn估计器。

第二个类中，它将符号特征名称序列（字符串strings）转化为scipy.sparse矩阵，使用哈希函数来计算对应每个名字的矩阵列。

从图像

该sklearn.feature\_extraction.image子模块收集实用程序从图像中提取特征。

|  |  |
| --- | --- |
| [feature\_extraction.image.img\_to\_graph](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.image.img_to_graph.html" \l "sklearn.feature_extraction.image.img_to_graph" \o "sklearn.feature_extraction.image.img_to_graph" \t "http://blog.csdn.net/u010859707/article/details/_blank)(img[, ...]) | 像素到像素梯度连接的图形 |
| [feature\_extraction.image.grid\_to\_graph](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.image.grid_to_graph.html" \l "sklearn.feature_extraction.image.grid_to_graph" \o "sklearn.feature_extraction.image.grid_to_graph" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n\_x, n\_y) | 像素到像素连接的图形 |
| [feature\_extraction.image.extract\_patches\_2d](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.image.extract_patches_2d.html" \l "sklearn.feature_extraction.image.extract_patches_2d" \o "sklearn.feature_extraction.image.extract_patches_2d" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 将2D图像重新整理成一组补丁 |
| [feature\_extraction.image.reconstruct\_from\_patches\_2d](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.image.reconstruct_from_patches_2d.html" \l "sklearn.feature_extraction.image.reconstruct_from_patches_2d" \o "sklearn.feature_extraction.image.reconstruct_from_patches_2d" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 从所有补丁重构图像 |
| [feature\_extraction.image.PatchExtractor](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.image.PatchExtractor.html" \l "sklearn.feature_extraction.image.PatchExtractor" \o "sklearn.feature_extraction.image.PatchExtractor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 从图像集中提取补丁 |

从文本

该sklearn.feature\_extraction.text子模块收集实用程序从文本文档建立特征向量。

|  |  |
| --- | --- |
| [feature\_extraction.text.CountVectorizer](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html" \l "sklearn.feature_extraction.text.CountVectorizer" \o "sklearn.feature_extraction.text.CountVectorizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 将文本文档的集合转换为令牌计数矩阵 |
| [feature\_extraction.text.HashingVectorizer](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html" \l "sklearn.feature_extraction.text.HashingVectorizer" \o "sklearn.feature_extraction.text.HashingVectorizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 将文本文档的集合转换为令牌发生的矩阵 |
| [feature\_extraction.text.TfidfTransformer](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html" \l "sklearn.feature_extraction.text.TfidfTransformer" \o "sklearn.feature_extraction.text.TfidfTransformer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 将计数矩阵转换为标准化的tf或tf-idf表示 |
| [feature\_extraction.text.TfidfVectorizer](http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html" \l "sklearn.feature_extraction.text.TfidfVectorizer" \o "sklearn.feature_extraction.text.TfidfVectorizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 将原始文档的集合转换为TF-IDF功能的矩阵 |

## sklearn.feature\_selection: Feature Selection（特征选择）

该sklearn.feature\_selection模块实现特征选择算法。它目前包括单变量筛选方法和递归特征消除算法。

类：

|  |  |
| --- | --- |
| [feature\_selection.GenericUnivariateSelect](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.GenericUnivariateSelect.html" \l "sklearn.feature_selection.GenericUnivariateSelect" \o "sklearn.feature_selection.GenericUnivariateSelect" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 具有可配置策略的单变量特征选择器 |
| [feature\_selection.SelectPercentile](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html" \l "sklearn.feature_selection.SelectPercentile" \o "sklearn.feature_selection.SelectPercentile" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 根据最高分数百分位数选择功能 |
| [feature\_selection.SelectKBest](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html" \l "sklearn.feature_selection.SelectKBest" \o "sklearn.feature_selection.SelectKBest" \t "http://blog.csdn.net/u010859707/article/details/_blank)([score\_func, k]) | 根据k最高分选择功能 |
| [feature\_selection.SelectFpr](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFpr.html" \l "sklearn.feature_selection.SelectFpr" \o "sklearn.feature_selection.SelectFpr" \t "http://blog.csdn.net/u010859707/article/details/_blank)([score\_func, alpha]) | 过滤器：根据FPR测试选择低于alpha的p值 |
| [feature\_selection.SelectFdr](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFdr.html" \l "sklearn.feature_selection.SelectFdr" \o "sklearn.feature_selection.SelectFdr" \t "http://blog.csdn.net/u010859707/article/details/_blank)([score\_func, alpha]) | 过滤器：为估计的错误发现率选择p值 |
| [feature\_selection.SelectFromModel](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html" \l "sklearn.feature_selection.SelectFromModel" \o "sklearn.feature_selection.SelectFromModel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator) | 元变压器，用于根据重要性权重选择特征 |
| [feature\_selection.SelectFwe](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFwe.html" \l "sklearn.feature_selection.SelectFwe" \o "sklearn.feature_selection.SelectFwe" \t "http://blog.csdn.net/u010859707/article/details/_blank)([score\_func, alpha]) | 过滤器：选择对应于同系误差率的p值 |
| [feature\_selection.RFE](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html" \l "sklearn.feature_selection.RFE" \o "sklearn.feature_selection.RFE" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator[, ...]) | 功能排序与递归功能消除 |
| [feature\_selection.RFECV](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html" \l "sklearn.feature_selection.RFECV" \o "sklearn.feature_selection.RFECV" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator[, step, ...]) | 功能排序与递归功能消除和交叉验证选择最佳数量的功能 |
| [feature\_selection.VarianceThreshold](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html" \l "sklearn.feature_selection.VarianceThreshold" \o "sklearn.feature_selection.VarianceThreshold" \t "http://blog.csdn.net/u010859707/article/details/_blank)([threshold]) | 功能选择器可删除所有低方差特征 |

RFE(Recursive feature elimination，递归特征选择)，不单独的检验某个变量的价值，而是将其聚集在一起检验。它的基本思想是，对于一个数量为d的特征的集合，他的所有的子集的个数是2的d次方减1（包含空集）。指定一个外部的学习算法，比如SVM之类的。通过该算法计算所有子集的validation error。选择error最小的那个子集作为所挑选的特征。

函数：

|  |  |
| --- | --- |
| [feature\_selection.chi2](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html" \l "sklearn.feature_selection.chi2" \o "sklearn.feature_selection.chi2" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y) | 计算每个非负特征和类之间的平方统计 |
| [feature\_selection.f\_classif](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html" \l "sklearn.feature_selection.f_classif" \o "sklearn.feature_selection.f_classif" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y) | 计算提供的样本的方差分析F值 |
| [feature\_selection.f\_regression](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html" \l "sklearn.feature_selection.f_regression" \o "sklearn.feature_selection.f_regression" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y[, center]) | 单变量线性回归测试 |
| [feature\_selection.mutual\_info\_classif](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html" \l "sklearn.feature_selection.mutual_info_classif" \o "sklearn.feature_selection.mutual_info_classif" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y) | 估计离散目标变量的互信息 |
| [feature\_selection.mutual\_info\_regression](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html" \l "sklearn.feature_selection.mutual_info_regression" \o "sklearn.feature_selection.mutual_info_regression" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y) | 估计连续目标变量的互信息 |

1. **sklearn.gaussian\_process:Gaussian Processes（高斯过程）**

该sklearn.gaussian\_process模块实现了基于高斯过程的回归和分类。

类：

|  |  |
| --- | --- |
| [gaussian\_process.GaussianProcessRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html" \l "sklearn.gaussian_process.GaussianProcessRegressor" \o "sklearn.gaussian_process.GaussianProcessRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 高斯过程回归（GPR） |
| [gaussian\_process.GaussianProcessClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html" \l "sklearn.gaussian_process.GaussianProcessClassifier" \o "sklearn.gaussian_process.GaussianProcessClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 基于拉普拉斯逼近的高斯过程分类（GPC） |

内核：

|  |  |
| --- | --- |
| [gaussian\_process.kernels.Kernel](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Kernel.html" \l "sklearn.gaussian_process.kernels.Kernel" \o "sklearn.gaussian_process.kernels.Kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 所有内核的基类 |
| [gaussian\_process.kernels.Sum](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Sum.html" \l "sklearn.gaussian_process.kernels.Sum" \o "sklearn.gaussian_process.kernels.Sum" \t "http://blog.csdn.net/u010859707/article/details/_blank)(k1, k2) | 两个内核k1和k2的和核k1 + k2 |
| [gaussian\_process.kernels.Product](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Product.html" \l "sklearn.gaussian_process.kernels.Product" \o "sklearn.gaussian_process.kernels.Product" \t "http://blog.csdn.net/u010859707/article/details/_blank)(k1, k2) | 两个内核k1和k2的产品内核k1\* k2 |
| [gaussian\_process.kernels.Exponentiation](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Exponentiation.html" \l "sklearn.gaussian_process.kernels.Exponentiation" \o "sklearn.gaussian_process.kernels.Exponentiation" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 通过给定指数来指定内核 |
| [gaussian\_process.kernels.ConstantKernel](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.ConstantKernel.html" \l "sklearn.gaussian_process.kernels.ConstantKernel" \o "sklearn.gaussian_process.kernels.ConstantKernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 恒定内核 |
| [gaussian\_process.kernels.WhiteKernel](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.WhiteKernel.html" \l "sklearn.gaussian_process.kernels.WhiteKernel" \o "sklearn.gaussian_process.kernels.WhiteKernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 白内核 |
| [gaussian\_process.kernels.RBF](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RBF.html" \l "sklearn.gaussian_process.kernels.RBF" \o "sklearn.gaussian_process.kernels.RBF" \t "http://blog.csdn.net/u010859707/article/details/_blank)([length\_scale, ...]) | 径向基函数核（又称平方指数核） |
| [gaussian\_process.kernels.Matern](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Matern.html" \l "sklearn.gaussian_process.kernels.Matern" \o "sklearn.gaussian_process.kernels.Matern" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | Matern 内核. |
| [gaussian\_process.kernels.RationalQuadratic](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RationalQuadratic.html" \l "sklearn.gaussian_process.kernels.RationalQuadratic" \o "sklearn.gaussian_process.kernels.RationalQuadratic" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 理性二次内核 |
| [gaussian\_process.kernels.ExpSineSquared](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.ExpSineSquared.html" \l "sklearn.gaussian_process.kernels.ExpSineSquared" \o "sklearn.gaussian_process.kernels.ExpSineSquared" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 正弦平方内核 |
| [gaussian\_process.kernels.DotProduct](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.DotProduct.html" \l "sklearn.gaussian_process.kernels.DotProduct" \o "sklearn.gaussian_process.kernels.DotProduct" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | Dot-Product内核 |
| [gaussian\_process.kernels.PairwiseKernel](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.PairwiseKernel.html" \l "sklearn.gaussian_process.kernels.PairwiseKernel" \o "sklearn.gaussian_process.kernels.PairwiseKernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 在sklearn.metrics.pairwise中的内核包装器 |
| [gaussian\_process.kernels.CompoundKernel](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.CompoundKernel.html" \l "sklearn.gaussian_process.kernels.CompoundKernel" \o "sklearn.gaussian_process.kernels.CompoundKernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(kernels) | 内核由一组其他内核组成 |
| [gaussian\_process.kernels.Hyperparameter](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Hyperparameter.html" \l "sklearn.gaussian_process.kernels.Hyperparameter" \o "sklearn.gaussian_process.kernels.Hyperparameter" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 内核超参数的指定形式为namedtuple |

sklearn.gaussian\_process.kernels.ConstantKernel可以用作核心产品的一部分，它划分其他因子（内核）的大小，或者作为一个总结内核的一部分修改高斯过程的平均值。

## sklearn.isotonic: Isotonic regression（保序回归）

保序回归的应用之一就是用来做统计推断，比如药量和毒性的关系，一般认为毒性随着药量是不减或者递增的关系，借此可以来估计最大药量。这种回归，是这一种单调函数的回归，回归模型中后一个x一定比前一个x大，也就是有序，具体的数学公式在上面两个网址中都有。保序回归并不需要制定的目标函数。

类：

|  |  |
| --- | --- |
| [isotonic.IsotonicRegression](http://scikit-learn.org/stable/modules/generated/sklearn.isotonic.IsotonicRegression.html" \l "sklearn.isotonic.IsotonicRegression" \o "sklearn.isotonic.IsotonicRegression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([y\_min, y\_max, ...]) | 保序回归模型 |

函数：

|  |  |
| --- | --- |
| [isotonic.isotonic\_regression](http://scikit-learn.org/stable/modules/generated/sklearn.isotonic.isotonic_regression.html" \l "sklearn.isotonic.isotonic_regression" \o "sklearn.isotonic.isotonic_regression" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y[, ...]) | 求解保序回归模型: |
| [isotonic.check\_increasing](http://scikit-learn.org/stable/modules/generated/sklearn.isotonic.check_increasing.html" \l "sklearn.isotonic.check_increasing" \o "sklearn.isotonic.check_increasing" \t "http://blog.csdn.net/u010859707/article/details/_blank)(x, y) | 确定y是否与x单调相关 |

## sklearn.kernel\_approximation Kernel Approximation（内核近似）

该sklearn.kernel\_approximation模块基于傅里叶变换实现几个近似核特征图。

类：

|  |  |
| --- | --- |
| [kernel\_approximation.AdditiveChi2Sampler](http://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.AdditiveChi2Sampler.html" \l "sklearn.kernel_approximation.AdditiveChi2Sampler" \o "sklearn.kernel_approximation.AdditiveChi2Sampler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 加性chi2核的近似特征图 |
| [kernel\_approximation.Nystroem](http://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.Nystroem.html" \l "sklearn.kernel_approximation.Nystroem" \o "sklearn.kernel_approximation.Nystroem" \t "http://blog.csdn.net/u010859707/article/details/_blank)([kernel, ...]) | 使用训练数据的子集近似一个内核映射 |
| [kernel\_approximation.RBFSampler](http://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.RBFSampler.html" \l "sklearn.kernel_approximation.RBFSampler" \o "sklearn.kernel_approximation.RBFSampler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([gamma, ...]) | 通过其傅立叶变换的Monte Carlo近似近似RBF核的特征图 |
| [kernel\_approximation.SkewedChi2Sampler](http://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.SkewedChi2Sampler.html" \l "sklearn.kernel_approximation.SkewedChi2Sampler" \o "sklearn.kernel_approximation.SkewedChi2Sampler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 通过其傅立叶变换的蒙特卡罗近似近似的“偏斜卡方”核的特征图 |

## sklearn.kernel\_ridge Kernel Ridge Regression（内核岭回归）

模块sklearn.kernel\_ridge实现内核脊回归。

类：

|  |  |
| --- | --- |
| [kernel\_ridge.KernelRidge](http://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html" \l "sklearn.kernel_ridge.KernelRidge" \o "sklearn.kernel_ridge.KernelRidge" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, kernel, ...]) | 内核岭回归 |

 Kernel Ridge Regression即使用核技巧的岭回归（L2正则线性回归），它的学习形式和SVR（support vector regression）相同，但是两者的损失函数不同。KRR有近似形式的解，并且在中度规模的数据时及其有效率，由于KRR没有参数稀疏化的性能，因此速度上要慢于SVR（它的损失函数有利于得到稀疏化的解）。

## 19 sklearn.linear\_model: Generalized Linear Models(广义线性模型）

## 该[sklearn.linear\_model](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.linear_model" \o "sklearn.linear_model" \t "http://blog.csdn.net/u010859707/article/details/_blank)模块实现广义线性模型。它包括利用最小角度回归和坐标下降计算的岭回归，贝叶斯回归，套索和弹性网估计。它还实现随机梯度下降相关算法。

|  |  |
| --- | --- |
| [linear\_model.ARDRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ARDRegression.html" \l "sklearn.linear_model.ARDRegression" \o "sklearn.linear_model.ARDRegression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_iter, tol, ...]) | 贝叶斯ARD回归 |
| [linear\_model.BayesianRidge](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.BayesianRidge.html" \l "sklearn.linear_model.BayesianRidge" \o "sklearn.linear_model.BayesianRidge" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_iter, tol, ...]) | 贝叶斯岭回归 |
| [linear\_model.ElasticNet](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet" \o "sklearn.linear_model.ElasticNet" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, l1\_ratio, ...]) | 线性回归与组合L1和L2先验作为正则化器 |
| [linear\_model.ElasticNetCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNetCV.html" \l "sklearn.linear_model.ElasticNetCV" \o "sklearn.linear_model.ElasticNetCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([l1\_ratio, eps, ...]) | 弹性网模型沿正则化路径迭代拟合 |
| [linear\_model.HuberRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.HuberRegressor.html" \l "sklearn.linear_model.HuberRegressor" \o "sklearn.linear_model.HuberRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([epsilon, ...]) | 线性回归模型，对离群值是robust |
| [linear\_model.Lars](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lars.html" \l "sklearn.linear_model.Lars" \o "sklearn.linear_model.Lars" \t "http://blog.csdn.net/u010859707/article/details/_blank)([fit\_intercept, verbose, ...]) | 最小角度回归模型 |
| [linear\_model.LarsCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LarsCV.html" \l "sklearn.linear_model.LarsCV" \o "sklearn.linear_model.LarsCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([fit\_intercept, ...]) | 交叉验证的最小二乘回归模型 |
| [linear\_model.Lasso](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso" \o "sklearn.linear_model.Lasso" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, fit\_intercept, ...]) | 线性模型训练用L1作为矫正器（又名拉索） |
| [linear\_model.LassoCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html" \l "sklearn.linear_model.LassoCV" \o "sklearn.linear_model.LassoCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([eps, n\_alphas, ...]) | 拉索线性模型，沿正则化路径迭代拟合 |
| [linear\_model.LassoLars](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoLars.html" \l "sklearn.linear_model.LassoLars" \o "sklearn.linear_model.LassoLars" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, ...]) | Lasso模型也适合最小角度回归 |
| [linear\_model.LassoLarsCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoLarsCV.html" \l "sklearn.linear_model.LassoLarsCV" \o "sklearn.linear_model.LassoLarsCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([fit\_intercept, ...]) | 使用LARS算法进行交叉验证的Lasso |
| [linear\_model.LassoLarsIC](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoLarsIC.html" \l "sklearn.linear_model.LassoLarsIC" \o "sklearn.linear_model.LassoLarsIC" \t "http://blog.csdn.net/u010859707/article/details/_blank)([criterion, ...]) | Lasso模型适合Lars使用BIC或AIC进行型号选择 |
| [linear\_model.LinearRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression" \o "sklearn.linear_model.LinearRegression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 普通最小二乘线性回归 |
| [linear\_model.LogisticRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression" \o "sklearn.linear_model.LogisticRegression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([penalty, ...]) | Logistic回归（又名logit，MaxEnt）分类器 |
| [linear\_model.LogisticRegressionCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html" \l "sklearn.linear_model.LogisticRegressionCV" \o "sklearn.linear_model.LogisticRegressionCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([Cs, ...]) | Logistic回归CV（又名logit，MaxEnt）分类器 |
| [linear\_model.MultiTaskLasso](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.MultiTaskLasso.html" \l "sklearn.linear_model.MultiTaskLasso" \o "sklearn.linear_model.MultiTaskLasso" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, ...]) | 用L1 / L2混合规范训练的多任务Lasso模型作为正则化器 |
| [linear\_model.MultiTaskElasticNet](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.MultiTaskElasticNet.html" \l "sklearn.linear_model.MultiTaskElasticNet" \o "sklearn.linear_model.MultiTaskElasticNet" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, ...]) | 用L1 / L2混合规范训练的多任务ElasticNet模型作为正则化程序 |
| [linear\_model.MultiTaskLassoCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.MultiTaskLassoCV.html" \l "sklearn.linear_model.MultiTaskLassoCV" \o "sklearn.linear_model.MultiTaskLassoCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([eps, ...]) | 多任务L1 / L2 Lasso内置交叉验证 |
| [linear\_model.MultiTaskElasticNetCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.MultiTaskElasticNetCV.html" \l "sklearn.linear_model.MultiTaskElasticNetCV" \o "sklearn.linear_model.MultiTaskElasticNetCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 多任务L1 / L2 ElasticNet内置交叉验证 |
| [linear\_model.OrthogonalMatchingPursuit](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.OrthogonalMatchingPursuit.html" \l "sklearn.linear_model.OrthogonalMatchingPursuit" \o "sklearn.linear_model.OrthogonalMatchingPursuit" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 正交匹配追踪模型（OMP） |
| [linear\_model.OrthogonalMatchingPursuitCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.OrthogonalMatchingPursuitCV.html" \l "sklearn.linear_model.OrthogonalMatchingPursuitCV" \o "sklearn.linear_model.OrthogonalMatchingPursuitCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 交叉验证的正交匹配追踪模型（OMP） |
| [linear\_model.PassiveAggressiveClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PassiveAggressiveClassifier.html" \l "sklearn.linear_model.PassiveAggressiveClassifier" \o "sklearn.linear_model.PassiveAggressiveClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 被动侵略分类器 |
| [linear\_model.PassiveAggressiveRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PassiveAggressiveRegressor.html" \l "sklearn.linear_model.PassiveAggressiveRegressor" \o "sklearn.linear_model.PassiveAggressiveRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([C, ...]) | 被动侵略者 |
| [linear\_model.Perceptron](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html" \l "sklearn.linear_model.Perceptron" \o "sklearn.linear_model.Perceptron" \t "http://blog.csdn.net/u010859707/article/details/_blank)([penalty, alpha, ...]) | 在“ [用户指南”中](http://scikit-learn.org/stable/modules/linear_model.html" \l "perceptron" \t "http://blog.csdn.net/u010859707/article/details/_blank)阅读更多内容。 |
| [linear\_model.RandomizedLasso](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RandomizedLasso.html" \l "sklearn.linear_model.RandomizedLasso" \o "sklearn.linear_model.RandomizedLasso" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, ...]) | 随机拉索 |
| [linear\_model.RandomizedLogisticRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RandomizedLogisticRegression.html" \l "sklearn.linear_model.RandomizedLogisticRegression" \o "sklearn.linear_model.RandomizedLogisticRegression" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 随机逻辑回归 |
| [linear\_model.RANSACRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html" \l "sklearn.linear_model.RANSACRegressor" \o "sklearn.linear_model.RANSACRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | RANSAC（RANdom SAmple Consensus）算法 |
| [linear\_model.Ridge](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html" \l "sklearn.linear_model.Ridge" \o "sklearn.linear_model.Ridge" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, fit\_intercept, ...]) | 具有l2正则化的线性最小二乘法 |
| [linear\_model.RidgeClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html" \l "sklearn.linear_model.RidgeClassifier" \o "sklearn.linear_model.RidgeClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, ...]) | 分类器使用Ridge回归 |
| [linear\_model.RidgeClassifierCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifierCV.html" \l "sklearn.linear_model.RidgeClassifierCV" \o "sklearn.linear_model.RidgeClassifierCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alphas, ...]) | 具有交叉验证的岭分类器 |
| [linear\_model.RidgeCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html" \l "sklearn.linear_model.RidgeCV" \o "sklearn.linear_model.RidgeCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alphas, ...]) | 里奇回归与内置交叉验证 |
| [linear\_model.SGDClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html" \l "sklearn.linear_model.SGDClassifier" \o "sklearn.linear_model.SGDClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([loss, penalty, ...]) | 线性分类器（SVM，逻辑回归，ao）与SGD训练 |
| [linear\_model.SGDRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html" \l "sklearn.linear_model.SGDRegressor" \o "sklearn.linear_model.SGDRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([loss, penalty, ...]) | 通过使用SGD最小化正则化经验损失拟合的线性模型 |
| [linear\_model.TheilSenRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.TheilSenRegressor.html" \l "sklearn.linear_model.TheilSenRegressor" \o "sklearn.linear_model.TheilSenRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | Theil-Sen估计：强大的多变量回归模型 |

从以上表格中我们可以具体看到每个类的具体功能：

Bayesian ARD regression（贝叶斯ARD回归）

首先使用ARD来拟合回归模型的权重。在高斯分布中回归模型的权重是假定的，还估计参数（分布的精确性）和α（噪声分布的精确性）。这个估计是通过一个迭代过程来完成的。

Bayesian ridge regression（贝叶斯岭回归）

拟合贝叶斯岭模型，优化正则化参数λ（权重的精度）和α（噪声的精密度）。

linear\_model.ElasticNet（线性回归与组合L1和L2先验作为正则化器）

结合L1和L2的先验正则化的线性回归。

linear\_model.ElasticNetCV（弹性网模型沿正则化路径迭代拟合）

正则化路径上迭代拟合的弹性网模型。

通过交叉验证选出最佳模型。

linear\_model.HuberRegressor（Huber 回归）

对异常值具有鲁棒性的线性回归模型。

linear\_model.Lars（最小角度回归模型）

最小角回归模型即LAR。

linear\_model.LarsCV（交叉验证的最小二乘回归模型）

交叉验证最小角度回归模型。

linear\_model.LassoCV（交叉验证的最小二乘回归模型）

正则化路径上迭代拟合的套索线性模型。

通过交叉验证选出最佳模型。

linear\_model.LassoLars（Lasso模型也适合最小角度回归）

最小角回归又名Lars套索模型拟合，

这是一个与之前的L1训练线性模型化。

linear\_model.RANSACRegressor（RANSAC（RANdom SAmple Consensus）算法）

RANSAC是一种迭代算法，是一个子集参数稳健的一个完整数据集。

linear\_model.Ridge（具有l2正则化的线性最小二乘法）

该模型解决了回归模型的损失函数是由线性函数和正则化最小二乘法L2范数给出这样一个问题。

linear\_model.RidgeClassifierCV（具有交叉验证的岭分类器）

是一种具有交叉验证的岭分类器。

默认情况下，它执行广义交叉验证，这是一种高效的一次性交叉验证形式。

linear\_model.RidgeCV（具有内建交叉验证的岭回归）

具有内建交叉验证的岭回归。

默认情况下，它执行广义交叉验证，这是一种高效的一次性交叉验证形式。

接下来介绍在这个类中要具体用到的函数表达以及起对应的函数的含义。

函数

|  |  |
| --- | --- |
| [linear\_model.lars\_path](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.lars_path.html" \l "sklearn.linear_model.lars_path" \o "sklearn.linear_model.lars_path" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y[, Xy, Gram, ...]) | 使用LARS算法计算最小角度回归或套索路径[1] |
| [linear\_model.lasso\_path](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.lasso_path.html" \l "sklearn.linear_model.lasso_path" \o "sklearn.linear_model.lasso_path" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y[, eps, ...]) | 计算具有坐标下降的Lasso路径 |
| [linear\_model.lasso\_stability\_path](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.lasso_stability_path.html" \l "sklearn.linear_model.lasso_stability_path" \o "sklearn.linear_model.lasso_stability_path" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y[, ...]) | 基于随机拉索估计的稳定性路径 |
| [linear\_model.logistic\_regression\_path](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.logistic_regression_path.html" \l "sklearn.linear_model.logistic_regression_path" \o "sklearn.linear_model.logistic_regression_path" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y) | 为正则化参数列表计算逻辑回归模型 |
| [linear\_model.orthogonal\_mp](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.orthogonal_mp.html" \l "sklearn.linear_model.orthogonal_mp" \o "sklearn.linear_model.orthogonal_mp" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y[, ...]) | 正交匹配追踪（OMP） |
| [linear\_model.orthogonal\_mp\_gram](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.orthogonal_mp_gram.html" \l "sklearn.linear_model.orthogonal_mp_gram" \o "sklearn.linear_model.orthogonal_mp_gram" \t "http://blog.csdn.net/u010859707/article/details/_blank)(Gram, Xy[, ...]) | 革命正交匹配追踪（OMP） |

## sklearn.manifold: Manifold Learning（歧管学习）

该[sklearn.manifold](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.manifold" \o "sklearn.manifold" \t "http://blog.csdn.net/u010859707/article/details/_blank)模块实现数据嵌入技术。

|  |  |
| --- | --- |
| [manifold.LocallyLinearEmbedding](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.LocallyLinearEmbedding.html" \l "sklearn.manifold.LocallyLinearEmbedding" \o "sklearn.manifold.LocallyLinearEmbedding" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 局部线性嵌入 |
| [manifold.Isomap](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html" \l "sklearn.manifold.Isomap" \o "sklearn.manifold.Isomap" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_neighbors, n\_components, ...]) | Isomap嵌入 |
| [manifold.MDS](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html" \l "sklearn.manifold.MDS" \o "sklearn.manifold.MDS" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, metric, n\_init, ...]) | 多维缩放 |
| [manifold.SpectralEmbedding](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.SpectralEmbedding.html" \l "sklearn.manifold.SpectralEmbedding" \o "sklearn.manifold.SpectralEmbedding" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 用于非线性维数降低的光谱嵌入 |
| [manifold.TSNE](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html" \l "sklearn.manifold.TSNE" \o "sklearn.manifold.TSNE" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, perplexity, ...]) | t分布随机相邻嵌入 |

manifold.Isomap实现的是Isomap嵌入和等距映射的非线性降维的功能，manifold.LocallyLinearEmbedding实现的局部线性嵌入的功能；[manifold.MDS](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html" \l "sklearn.manifold.MDS" \o "sklearn.manifold.MDS" \t "http://blog.csdn.net/u010859707/article/details/_blank)实现的是多维缩放；[manifold.SpectralEmbedding](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.SpectralEmbedding.html" \l "sklearn.manifold.SpectralEmbedding" \o "sklearn.manifold.SpectralEmbedding" \t "http://blog.csdn.net/u010859707/article/details/_blank)用于非线性降维的光谱嵌入，形成一个由指定函数赋予的亲和矩阵，并将光谱分解应用到相应的函数中。manifold.TSNE实现的是t分布随机相邻嵌入的功能，t-SNE是一个将高维数据可视化的一个工具，它将数据点之间的相似性转换为联合概率，并将低维嵌入和多维数据的联合概率之间的Kullback-Leibler差异最小化。T-SNE有非凸的成本函数，即不同的初始化，我们可以得到不同的结果。

这是在这些类中会用到的函数：

函数

|  |  |
| --- | --- |
| [manifold.locally\_linear\_embedding](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.locally_linear_embedding.html" \l "sklearn.manifold.locally_linear_embedding" \o "sklearn.manifold.locally_linear_embedding" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, ...[, ...]) | 对数据进行局部线性嵌入分析 |
| [manifold.spectral\_embedding](http://scikit-learn.org/stable/modules/generated/sklearn.manifold.spectral_embedding.html" \l "sklearn.manifold.spectral_embedding" \o "sklearn.manifold.spectral_embedding" \t "http://blog.csdn.net/u010859707/article/details/_blank)(adjacency[, ...]) | 将样本投影在拉普拉斯算子的第一个特征向量上 |

## sklearn.metrics: Metrics（指标）

该[sklearn.metrics](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.metrics" \o "sklearn.metrics" \t "http://blog.csdn.net/u010859707/article/details/_blank)模块包括分数函数，性能度量和成对度量和距离计算。

Model Selection Interface（选型接口）

|  |  |
| --- | --- |
| [metrics.make\_scorer](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.make_scorer.html" \l "sklearn.metrics.make_scorer" \o "sklearn.metrics.make_scorer" \t "http://blog.csdn.net/u010859707/article/details/_blank)(score\_func[, ...]) | 从表现指标或损失函数中取得记分员 |
| [metrics.get\_scorer](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.get_scorer.html" \l "sklearn.metrics.get_scorer" \o "sklearn.metrics.get_scorer" \t "http://blog.csdn.net/u010859707/article/details/_blank)(scoring) | 从一个字符串得到一个分数 |

Classification metrics（分类度量）

|  |  |
| --- | --- |
| [metrics.accuracy\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html" \l "sklearn.metrics.accuracy_score" \o "sklearn.metrics.accuracy_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 准确度分级得分 |
| [metrics.auc](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html" \l "sklearn.metrics.auc" \o "sklearn.metrics.auc" \t "http://blog.csdn.net/u010859707/article/details/_blank)(x, y[, reorder]) | 曲线下的计算面积（AUC）使用梯形规则 |
| [metrics.average\_precision\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html" \l "sklearn.metrics.average_precision_score" \o "sklearn.metrics.average_precision_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_score) | 从预测分数计算平均精度（AP） |
| [metrics.brier\_score\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.brier_score_loss.html" \l "sklearn.metrics.brier_score_loss" \o "sklearn.metrics.brier_score_loss" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_prob[, ...]) | 计算Brier分数 |
| [metrics.classification\_report](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html" \l "sklearn.metrics.classification_report" \o "sklearn.metrics.classification_report" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred) | 构建一个显示主要分类指标的文本报告 |
| [metrics.cohen\_kappa\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html" \l "sklearn.metrics.cohen_kappa_score" \o "sklearn.metrics.cohen_kappa_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y1, y2[, labels, ...]) | 科恩的kappa：衡量标注者间协议的统计 |
| [metrics.confusion\_matrix](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html" \l "sklearn.metrics.confusion_matrix" \o "sklearn.metrics.confusion_matrix" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 计算混淆矩阵来评估分类的准确性 |
| [metrics.f1\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html" \l "sklearn.metrics.f1_score" \o "sklearn.metrics.f1_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, labels, ...]) | 计算F1分数，也称为平衡F分数或F度量 |
| [metrics.fbeta\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.fbeta_score.html" \l "sklearn.metrics.fbeta_score" \o "sklearn.metrics.fbeta_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred, beta[, ...]) | 计算F-beta分数 |
| [metrics.hamming\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html" \l "sklearn.metrics.hamming_loss" \o "sklearn.metrics.hamming_loss" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 计算平均汉明损失 |
| [metrics.hinge\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.hinge_loss.html" \l "sklearn.metrics.hinge_loss" \o "sklearn.metrics.hinge_loss" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, pred\_decision[, ...]) | 平均铰链损失（非正规化） |
| [metrics.jaccard\_similarity\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_similarity_score.html" \l "sklearn.metrics.jaccard_similarity_score" \o "sklearn.metrics.jaccard_similarity_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred) | Jaccard相似系数得分 |
| [metrics.log\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html" \l "sklearn.metrics.log_loss" \o "sklearn.metrics.log_loss" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, eps, ...]) | 对数损失，又称物流损失或交叉熵损失 |
| [metrics.matthews\_corrcoef](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html" \l "sklearn.metrics.matthews_corrcoef" \o "sklearn.metrics.matthews_corrcoef" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 计算二进制类的马修斯相关系数（MCC） |
| [metrics.precision\_recall\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html" \l "sklearn.metrics.precision_recall_curve" \o "sklearn.metrics.precision_recall_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, ...) | 计算不同概率阈值的 precision-recall 对 |
| [metrics.precision\_recall\_fscore\_support](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html" \l "sklearn.metrics.precision_recall_fscore_support" \o "sklearn.metrics.precision_recall_fscore_support" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 计算每个课程的precision，recall，F度量和支持 |
| [metrics.precision\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html" \l "sklearn.metrics.precision_score" \o "sklearn.metrics.precision_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 计算precision |
| [metrics.recall\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html" \l "sklearn.metrics.recall_score" \o "sklearn.metrics.recall_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 计算recall |
| [metrics.roc\_auc\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html" \l "sklearn.metrics.roc_auc_score" \o "sklearn.metrics.roc_auc_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_score[, ...]) | 曲线下的计算面积（AUC）来自预测分数 |
| [metrics.roc\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html" \l "sklearn.metrics.roc_curve" \o "sklearn.metrics.roc_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_score[, ...]) | 计算接收器工作特性（ROC） |
| [metrics.zero\_one\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.zero_one_loss.html" \l "sklearn.metrics.zero_one_loss" \o "sklearn.metrics.zero_one_loss" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 零分类损失 |

Regression metrics(回归指标)

|  |  |
| --- | --- |
| [metrics.explained\_variance\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.explained_variance_score.html" \l "sklearn.metrics.explained_variance_score" \o "sklearn.metrics.explained_variance_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred) | 解释方差回归分数函数 |
| [metrics.mean\_absolute\_error](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html" \l "sklearn.metrics.mean_absolute_error" \o "sklearn.metrics.mean_absolute_error" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred) | 平均绝对误差回归损失 |
| [metrics.mean\_squared\_error](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html" \l "sklearn.metrics.mean_squared_error" \o "sklearn.metrics.mean_squared_error" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | 均方误差回归损失 |
| [metrics.median\_absolute\_error](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.median_absolute_error.html" \l "sklearn.metrics.median_absolute_error" \o "sklearn.metrics.median_absolute_error" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred) | 中值绝对误差回归损失 |
| [metrics.r2\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html" \l "sklearn.metrics.r2_score" \o "sklearn.metrics.r2_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_pred[, ...]) | R^2（测定系数）回归分数函数 |

Multilabel ranking metrics(多标签排名指标)

|  |  |
| --- | --- |
| [metrics.coverage\_error](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.coverage_error.html" \l "sklearn.metrics.coverage_error" \o "sklearn.metrics.coverage_error" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_score[, ...]) | 覆盖误差测量 |
| [metrics.label\_ranking\_average\_precision\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.label_ranking_average_precision_score.html" \l "sklearn.metrics.label_ranking_average_precision_score" \o "sklearn.metrics.label_ranking_average_precision_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 计算基于排名的平均精度 |
| [metrics.label\_ranking\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.label_ranking_loss.html" \l "sklearn.metrics.label_ranking_loss" \o "sklearn.metrics.label_ranking_loss" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y\_true, y\_score) | 计算排名损失量 |

Clustering metrics(聚类指标)

该[sklearn.metrics.cluster](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.metrics.cluster" \o "sklearn.metrics.cluster" \t "http://blog.csdn.net/u010859707/article/details/_blank)子模块包含了聚类分析的结果评价指标。有两种形式的评估：

（1）监督，它为每个样本使用地面真值类别值。

（2）无监督，不对和衡量模型本身的“质量”。

|  |  |
| --- | --- |
| [metrics.adjusted\_mutual\_info\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html" \l "sklearn.metrics.adjusted_mutual_info_score" \o "sklearn.metrics.adjusted_mutual_info_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 两个集群之间调整的相互信息 |
| [metrics.adjusted\_rand\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html" \l "sklearn.metrics.adjusted_rand_score" \o "sklearn.metrics.adjusted_rand_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels\_true, ...) | 兰德指数调整机会 |
| [metrics.calinski\_harabaz\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabaz_score.html" \l "sklearn.metrics.calinski_harabaz_score" \o "sklearn.metrics.calinski_harabaz_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, labels) | 计算Calinski和Harabaz得分 |
| [metrics.completeness\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.completeness_score.html" \l "sklearn.metrics.completeness_score" \o "sklearn.metrics.completeness_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels\_true, ...) | 给定一个地面真相的集群标签的完整度量 |
| [metrics.fowlkes\_mallows\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.fowlkes_mallows_score.html" \l "sklearn.metrics.fowlkes_mallows_score" \o "sklearn.metrics.fowlkes_mallows_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels\_true, ...) | 测量一组点的两个聚类的相似度 |
| [metrics.homogeneity\_completeness\_v\_measure](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_completeness_v_measure.html" \l "sklearn.metrics.homogeneity_completeness_v_measure" \o "sklearn.metrics.homogeneity_completeness_v_measure" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 一次计算同质性和完整性和V-Measure分数 |
| [metrics.homogeneity\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_score.html" \l "sklearn.metrics.homogeneity_score" \o "sklearn.metrics.homogeneity_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels\_true, ...) | 给出了一个地面事实的集群标签的均匀性度量 |
| [metrics.mutual\_info\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html" \l "sklearn.metrics.mutual_info_score" \o "sklearn.metrics.mutual_info_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels\_true, ...) | 两个集群之间的相互信息 |
| [metrics.normalized\_mutual\_info\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html" \l "sklearn.metrics.normalized_mutual_info_score" \o "sklearn.metrics.normalized_mutual_info_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 两个集群之间的归一化互信息 |
| [metrics.silhouette\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html" \l "sklearn.metrics.silhouette_score" \o "sklearn.metrics.silhouette_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, labels[, ...]) | 计算所有样本的平均轮廓系数 |
| [metrics.silhouette\_samples](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_samples.html" \l "sklearn.metrics.silhouette_samples" \o "sklearn.metrics.silhouette_samples" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, labels[, metric]) | 计算每个样本的剪影系数 |
| [metrics.v\_measure\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.v_measure_score.html" \l "sklearn.metrics.v_measure_score" \o "sklearn.metrics.v_measure_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels\_true, labels\_pred) | V-measure集群标签给出了一个基本的真相 |

Biclustering metrics(二聚体指标)

|  |  |
| --- | --- |
| [metrics.consensus\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.consensus_score.html" \l "sklearn.metrics.consensus_score" \o "sklearn.metrics.consensus_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(a, b[, similarity]) | 两组双核的相似性 |

Pairwise metrics(成对指标)

|  |  |
| --- | --- |
| [metrics.pairwise.additive\_chi2\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.additive_chi2_kernel.html" \l "sklearn.metrics.pairwise.additive_chi2_kernel" \o "sklearn.metrics.pairwise.additive_chi2_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y]) | 计算X和Y中观测值之间的加性卡方核 |
| [metrics.pairwise.chi2\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.chi2_kernel.html" \l "sklearn.metrics.pairwise.chi2_kernel" \o "sklearn.metrics.pairwise.chi2_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, gamma]) | 计算指数卡方核X和Y |
| [metrics.pairwise.distance\_metrics](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.distance_metrics.html" \l "sklearn.metrics.pairwise.distance_metrics" \o "sklearn.metrics.pairwise.distance_metrics" \t "http://blog.csdn.net/u010859707/article/details/_blank)() | pairwise\_distances的有效指标 |
| [metrics.pairwise.euclidean\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.euclidean_distances.html" \l "sklearn.metrics.pairwise.euclidean_distances" \o "sklearn.metrics.pairwise.euclidean_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 考虑X（和Y = X）的行作为向量，计算每对向量之间的距离矩阵 |
| [metrics.pairwise.kernel\_metrics](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.kernel_metrics.html" \l "sklearn.metrics.pairwise.kernel_metrics" \o "sklearn.metrics.pairwise.kernel_metrics" \t "http://blog.csdn.net/u010859707/article/details/_blank)() | pairwise\_kernels的有效指标 |
| [metrics.pairwise.linear\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.linear_kernel.html" \l "sklearn.metrics.pairwise.linear_kernel" \o "sklearn.metrics.pairwise.linear_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y]) | 计算X和Y之间的线性内核 |
| [metrics.pairwise.manhattan\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.manhattan_distances.html" \l "sklearn.metrics.pairwise.manhattan_distances" \o "sklearn.metrics.pairwise.manhattan_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 计算X和Y中向量之间的L1距离 |
| [metrics.pairwise.pairwise\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.pairwise_distances.html" \l "sklearn.metrics.pairwise.pairwise_distances" \o "sklearn.metrics.pairwise.pairwise_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 从矢量数组X和可选Y计算距离矩阵 |
| [metrics.pairwise.pairwise\_kernels](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.pairwise_kernels.html" \l "sklearn.metrics.pairwise.pairwise_kernels" \o "sklearn.metrics.pairwise.pairwise_kernels" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 计算阵列X和可选阵列Y之间的内核 |
| [metrics.pairwise.polynomial\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.polynomial_kernel.html" \l "sklearn.metrics.pairwise.polynomial_kernel" \o "sklearn.metrics.pairwise.polynomial_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 计算X和Y之间的多项式内核 |
| [metrics.pairwise.rbf\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.rbf_kernel.html" \l "sklearn.metrics.pairwise.rbf_kernel" \o "sklearn.metrics.pairwise.rbf_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, gamma]) | 计算X和Y之间的rbf（高斯）内核 |
| [metrics.pairwise.sigmoid\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.sigmoid_kernel.html" \l "sklearn.metrics.pairwise.sigmoid_kernel" \o "sklearn.metrics.pairwise.sigmoid_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 计算X和Y之间的S形内核 |
| [metrics.pairwise.cosine\_similarity](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html" \l "sklearn.metrics.pairwise.cosine_similarity" \o "sklearn.metrics.pairwise.cosine_similarity" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, ...]) | 计算X和Y中样本之间的余弦相似度 |
| [metrics.pairwise.cosine\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_distances.html" \l "sklearn.metrics.pairwise.cosine_distances" \o "sklearn.metrics.pairwise.cosine_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y]) | 计算X和Y中样本之间的余弦距离 |
| [metrics.pairwise.laplacian\_kernel](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.laplacian_kernel.html" \l "sklearn.metrics.pairwise.laplacian_kernel" \o "sklearn.metrics.pairwise.laplacian_kernel" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, gamma]) | 计算X和Y之间的拉普拉斯核 |
| [metrics.pairwise\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise_distances.html" \l "sklearn.metrics.pairwise_distances" \o "sklearn.metrics.pairwise_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, Y, metric, ...]) | 从矢量数组X和可选Y计算距离矩阵 |
| [metrics.pairwise\_distances\_argmin](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise_distances_argmin.html" \l "sklearn.metrics.pairwise_distances_argmin" \o "sklearn.metrics.pairwise_distances_argmin" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, Y[, ...]) | 计算一点与一组点之间的最小距离 |
| [metrics.pairwise\_distances\_argmin\_min](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise_distances_argmin_min.html" \l "sklearn.metrics.pairwise_distances_argmin_min" \o "sklearn.metrics.pairwise_distances_argmin_min" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, Y) | 计算一点与一组点之间的最小距离 |
| [metrics.pairwise.paired\_euclidean\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.paired_euclidean_distances.html" \l "sklearn.metrics.pairwise.paired_euclidean_distances" \o "sklearn.metrics.pairwise.paired_euclidean_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, Y) | 计算X与Y之间的配对欧氏距离 |
| [metrics.pairwise.paired\_manhattan\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.paired_manhattan_distances.html" \l "sklearn.metrics.pairwise.paired_manhattan_distances" \o "sklearn.metrics.pairwise.paired_manhattan_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, Y) | 计算X和Y中向量之间的L1距离 |
| [metrics.pairwise.paired\_cosine\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.paired_cosine_distances.html" \l "sklearn.metrics.pairwise.paired_cosine_distances" \o "sklearn.metrics.pairwise.paired_cosine_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, Y) | 计算X和Y之间的配对余弦距离 |
| [metrics.pairwise.paired\_distances](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.paired_distances.html" \l "sklearn.metrics.pairwise.paired_distances" \o "sklearn.metrics.pairwise.paired_distances" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, Y[, metric]) | 计算X和Y之间的配对距离 |

## sklearn.mixture: Gaussian Mixture Models（高斯混合模型）

该[sklearn.mixture](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.mixture" \o "sklearn.mixture" \t "http://blog.csdn.net/u010859707/article/details/_blank)模块实现混合建模算法。

|  |  |
| --- | --- |
| [mixture.GaussianMixture](http://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html" \l "sklearn.mixture.GaussianMixture" \o "sklearn.mixture.GaussianMixture" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 高斯混合 |
| [mixture.BayesianGaussianMixture](http://scikit-learn.org/stable/modules/generated/sklearn.mixture.BayesianGaussianMixture.html" \l "sklearn.mixture.BayesianGaussianMixture" \o "sklearn.mixture.BayesianGaussianMixture" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 高斯混合变分贝叶斯估计 |

[mixture.BayesianGaussianMixture](http://scikit-learn.org/stable/modules/generated/sklearn.mixture.BayesianGaussianMixture.html" \l "sklearn.mixture.BayesianGaussianMixture" \o "sklearn.mixture.BayesianGaussianMixture" \t "http://blog.csdn.net/u010859707/article/details/_blank)（高斯混合变分贝叶斯估计），这个类允许推断出一个近似的后验分布的高斯混合分布参数，且可以从数据中推断出有效组件的数量。这个类实现了两种先验的权重分配：一个与Dirichlet过程Dirchlet分布无限混合模型的有限混合模型。在实践中，Dirichlet过程推理算法近似，采用截尾分布与一个固定的最大数量的组件（称为“破局代表”），然而实际使用的组件数量几乎总是取决于数据。

Gaussian Mixture.（高斯混合）可表示高斯混合模型概率分布。而且这个类允许估计参数(一个高斯混合分布参数)。

## sklearn.model\_selection: Model Selection（模型选择）

### Splitter Classes（分割器类）

|  |  |
| --- | --- |
| [model\_selection.KFold](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html" \l "sklearn.model_selection.KFold" \o "sklearn.model_selection.KFold" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_splits, shuffle, ...]) | K-折叠交叉验证器 |
| [model\_selection.GroupKFold](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html" \l "sklearn.model_selection.GroupKFold" \o "sklearn.model_selection.GroupKFold" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_splits]) | 具有非重叠组的K-fold迭代器变体 |
| [model\_selection.StratifiedKFold](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html" \l "sklearn.model_selection.StratifiedKFold" \o "sklearn.model_selection.StratifiedKFold" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_splits, ...]) | 分层K-折叠交叉验证器 |
| [model\_selection.LeaveOneGroupOut](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.LeaveOneGroupOut.html" \l "sklearn.model_selection.LeaveOneGroupOut" \o "sklearn.model_selection.LeaveOneGroupOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)() | 离开一组交叉验证器 |
| [model\_selection.LeavePGroupsOut](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.LeavePGroupsOut.html" \l "sklearn.model_selection.LeavePGroupsOut" \o "sklearn.model_selection.LeavePGroupsOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n\_groups) | 离开P组交叉验证器 |
| [model\_selection.LeaveOneOut](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.LeaveOneOut.html" \l "sklearn.model_selection.LeaveOneOut" \o "sklearn.model_selection.LeaveOneOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)() | 一次性交叉验证器 |
| [model\_selection.LeavePOut](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.LeavePOut.html" \l "sklearn.model_selection.LeavePOut" \o "sklearn.model_selection.LeavePOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)(p) | Leave-P-Out交叉验证器 |
| [model\_selection.ShuffleSplit](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ShuffleSplit.html" \l "sklearn.model_selection.ShuffleSplit" \o "sklearn.model_selection.ShuffleSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_splits, ...]) | 随机置换交叉验证器 |
| [model\_selection.GroupShuffleSplit](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupShuffleSplit.html" \l "sklearn.model_selection.GroupShuffleSplit" \o "sklearn.model_selection.GroupShuffleSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 随机组 - 交叉验证迭代器 |
| [model\_selection.StratifiedShuffleSplit](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html" \l "sklearn.model_selection.StratifiedShuffleSplit" \o "sklearn.model_selection.StratifiedShuffleSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 分层ShuffleSplit交叉验证器 |
| [model\_selection.PredefinedSplit](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.PredefinedSplit.html" \l "sklearn.model_selection.PredefinedSplit" \o "sklearn.model_selection.PredefinedSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)(test\_fold) | 预定义分裂交叉验证器 |
| [model\_selection.TimeSeriesSplit](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html" \l "sklearn.model_selection.TimeSeriesSplit" \o "sklearn.model_selection.TimeSeriesSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_splits]) | 时间序列交叉验证器 |

model\_selection.GroupKFold（K-折叠交叉验证器），代表着k的迭代器的变异与非重叠的组。同一组不会出现在两个不同的折叠中（不同的组的数目必须至少等于折叠的数目）。一般来说，不同群体的数量与折叠的数目大致上相同的。

model\_selection.GroupShuffleSplit（随机组 - 交叉验证迭代器），根据第三方提供组提供随机的火车/测试指标拆分数据，这组信息可以用来对样品的任意特定领域层编码为整数。

Leavepgroupsout和GroupShuffleSplit之间的区别是，前者的生成将使用所有大小P独特的组的子集，而groupshufflesplit生成用户确定数量的随机试验分裂，每一个用户确定独特的组分数。model\_selection.KFold提供列车/测试指标，以分割列车/测试集中的数据。将数据集分割成k连续折叠（默认情况下无需重排）。然后每一次折叠作为验证，K - 1剩余折叠形成训练集。model\_selection.LeaveOneOut（一次性交叉验证器），提供列车/测试指标，以分割列车/测试集中的数据。每个样本作为测试集（单件）使用一次，其余的样本形成训练集。由于大量测试集（与样本数相同），导致这种交叉验证方法可能非常昂贵。所以针对大数据集，应支持Kfold，ShuffleSplit或stratifiedkfold。model\_selection.PredefinedSplit（预定义分裂交叉验证器），根据预定义的方案将数据分割成训练/测试集折叠。每个样品可以分配给

一个测试集折，所指定的用户通过test\_fold参数。

Splitter Functions（分割函数）

|  |  |
| --- | --- |
| [model\_selection.train\_test\_split](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html" \l "sklearn.model_selection.train_test_split" \o "sklearn.model_selection.train_test_split" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*arrays, ...) | 将阵列或矩阵拆分成随机列和测试子集 |
| [model\_selection.check\_cv](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.check_cv.html" \l "sklearn.model_selection.check_cv" \o "sklearn.model_selection.check_cv" \t "http://blog.csdn.net/u010859707/article/details/_blank)([cv, y, classifier]) | 用于构建交叉验证器的输入检查器实用程序 |

Hyper-parameter optimizers（超参数优化）

|  |  |  |  |
| --- | --- | --- | --- |
| [model\_selection.GridSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html" \l "sklearn.model_selection.GridSearchCV" \o "sklearn.model_selection.GridSearchCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, ...) | | 对估计器的指定参数值进行详尽搜索 | |
| [model\_selection.RandomizedSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html" \l "sklearn.model_selection.RandomizedSearchCV" \o "sklearn.model_selection.RandomizedSearchCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...[, ...]) | | 随机搜索超参数 | |
| [model\_selection.ParameterGrid](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ParameterGrid.html" \l "sklearn.model_selection.ParameterGrid" \o "sklearn.model_selection.ParameterGrid" \t "http://blog.csdn.net/u010859707/article/details/_blank)(param\_grid) | | 每个参数的网格具有离散数量的值 | |
| [model\_selection.ParameterSampler](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ParameterSampler.html" \l "sklearn.model_selection.ParameterSampler" \o "sklearn.model_selection.ParameterSampler" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...[, ...]) | | 发电机对从给定分布采样的参数 | |
| [model\_selection.fit\_grid\_point](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.fit_grid_point.html" \l "sklearn.model_selection.fit_grid_point" \o "sklearn.model_selection.fit_grid_point" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y, ...[, ...]) | 适合一组参数 | |

 model\_selection.ParameterSampler（发电机对从给定分布采样的参数），非确定性的随机组合进行超参数搜索。如果所有的参数列为一个列表，进行无需更换的采样。如果至少有一个参数为一个，则进行抽样。

Model validation（模型验证）

|  |  |
| --- | --- |
| [model\_selection.cross\_val\_score](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html" \l "sklearn.model_selection.cross_val_score" \o "sklearn.model_selection.cross_val_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, X) | 通过交叉验证评估分数 |
| [model\_selection.cross\_val\_predict](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_predict.html" \l "sklearn.model_selection.cross_val_predict" \o "sklearn.model_selection.cross_val_predict" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, X) | 为每个输入数据点生成交叉验证的估计 |
| [model\_selection.permutation\_test\_score](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.permutation_test_score.html" \l "sklearn.model_selection.permutation_test_score" \o "sklearn.model_selection.permutation_test_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 评估具有置换的交叉验证分数的意义 |
| [model\_selection.learning\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html" \l "sklearn.model_selection.learning_curve" \o "sklearn.model_selection.learning_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, X, y) | 学习曲线 |
| [model\_selection.validation\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.validation_curve.html" \l "sklearn.model_selection.validation_curve" \o "sklearn.model_selection.validation_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, ...) | 验证曲线 |

 model\_selection.learning\_curve（学习曲线），确定交叉验证的训练组和测试分数为不同的训练集大小。交叉验证生成器在训练和测试数据中分割整个数据集k次，训练集的不同大小的子集将被用来训练估计量和每个训练子集的大小和测试集将被计算成分数，然后，每个训练子集的大小都将在所有K值上计算平均得分。model\_selection.validation\_curve（验证曲线），确定不同参数值的训练和测试成绩。并且计算具有指定参数不同值的估计值。这类似于单参数的网络搜索。然而，这也将计算训练成绩，结果仅仅是一个用于绘图的工具。

## sklearn.multiclass: Multiclass and multilabel classification（多类和多标签分类）

该模块实现了多类学习算法：

. one-vs-the-rest / one-vs-all

. one-vs-one

. 纠错输出代码

该模块中提供的估计量是元估计器：它们需要在其构造函数中提供基本估计器。例如，可以使用这些估计器将二进制分类器或回归器转换为多类分类器。也可以将这些估计器与多类估计器一起使用，希望它们的准确性或运行时性能得到改善。

scikit-learn中的所有分类器实现多类分类; 您只需要使用此模块即可尝试使用自定义多类策略。

一对一的元分类器也实现了一个predict\_proba方法，只要这种方法由基类分类器实现即可。该方法在单个标签和多重标签的情况下返回类成员资格的概率。注意，在多重标签的情况下，概率是给定样本落在给定类中的边际概率。因此，在多标签情况下，这些概率在一个给定样本的所有可能的标签的总和不会和为1，因为他们在单个标签的情况下做的。

|  |  |
| --- | --- |
| [multiclass.OneVsRestClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html" \l "sklearn.multiclass.OneVsRestClassifier" \o "sklearn.multiclass.OneVsRestClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator[, ...]) | One-vs-the-rest (OvR) 多类/多标签策略 |
| [multiclass.OneVsOneClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsOneClassifier.html" \l "sklearn.multiclass.OneVsOneClassifier" \o "sklearn.multiclass.OneVsOneClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator[, ...]) | One-vs-one 多类策略 |
| [multiclass.OutputCodeClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OutputCodeClassifier.html" \l "sklearn.multiclass.OutputCodeClassifier" \o "sklearn.multiclass.OutputCodeClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator[, ...]) | （错误校正）输出代码多类策略 |

 multiclass.OneVsRestClassifier（One-vs-the-rest (OvR) 多类/多标签策略），这一策略也被称为一对一，这种策略包括每个类拟合一个分类器。对于每个分类器，类与所有其他类相匹配。此外，其计算效率（只n\_classes分类器需要），这种方法的一个优点是它的可解释性。因为每个类由一个和一个表示分类器只能通过检查相应的分类器来获得关于类的知识。这是最常用的多类分类策略，是一个公平的默认选择。

multiclass.OneVsOneClassifier（One-vs-one 多类策略），这一策略也被称为一对一，这种策略包括每个类拟合一个分类器。对于每个分类器，类与所有其他类相匹配。此外，其计算效率（只n\_classes分类器需要），这种方法的一个优点是它的可解释性。因为每个类由一个和一个表示分类器只能通过检查相应的分类器来获得关于类的知识。这是最常用的多类分类策略，是一个公平的默认选择。

multiclass.OutputCodeClassifier（（错误校正）输出代码多类策略），输出基于代码的策略包括在代表每个类的二进制码（0和1数组）。在恰当的时间，一个二进制分类器每点在书上的代码安装。在预测时，该分类方法在班级空间和接近点类新项目选择要点。这些主要的优势策略是量词的使用数量可以由用户控制，要么压缩模型（0 < code\_size＜1）或使模型更强大的错误（code\_size > 1）。

## sklearn.multioutput: Multioutput regression and classification（多输出回归和分类)

该模块实现多输出回归和分类。

该模块中提供的估计量是元估计器：它们需要在其构造函数中提供基本估计器。元估计器将单输出估计器扩展到多输出估计器。

|  |  |
| --- | --- |
| [multioutput.MultiOutputRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html" \l "sklearn.multioutput.MultiOutputRegressor" \o "sklearn.multioutput.MultiOutputRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator) | 多目标回归 |
| [multioutput.MultiOutputClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html" \l "sklearn.multioutput.MultiOutputClassifier" \o "sklearn.multioutput.MultiOutputClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator) | 多目标分类 |
| multioutput.ClassifierChain | 将二进制分类器排列成一个多标签的模型链 |

multioutput.ClassifierChain（将二进制分类器排列成一个多标签的模型

链。），每个模型使用链提供的所有可用特性加上链中较早的模型的预测，按照链指定的顺序进行预测。multioutput.MultiOutputRegressor（多目标回归），这个战略包括每一个自变量拟合目标。这是本身不支持多目标回归的一个延长的解释变量，这样一个简单的策略。multioutput.MultiOutputClassifier（多目标分类），该策略包括每个目标分别拟合一个分类器。这是延长分类器不原生支持多目标分类的一个简单的策略。

## sklearn.naive\_bayes: Naive Bayes（朴素贝叶斯）

该[sklearn.naive\_bayes](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.naive_bayes" \o "sklearn.naive_bayes" \t "http://blog.csdn.net/u010859707/article/details/_blank)模块实现朴素贝叶斯算法。这些是基于应用贝叶斯定理与强（天真）特征独立假设的监督学习方法。

|  |  |
| --- | --- |
| [naive\_bayes.GaussianNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html" \l "sklearn.naive_bayes.GaussianNB" \o "sklearn.naive_bayes.GaussianNB" \t "http://blog.csdn.net/u010859707/article/details/_blank)([priors]) | 高斯朴素贝叶斯（GaussianNB） |
| [naive\_bayes.MultinomialNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html" \l "sklearn.naive_bayes.MultinomialNB" \o "sklearn.naive_bayes.MultinomialNB" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, ...]) | 朴素贝叶斯分类器多项式模型 |
| [naive\_bayes.BernoulliNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html" \l "sklearn.naive_bayes.BernoulliNB" \o "sklearn.naive_bayes.BernoulliNB" \t "http://blog.csdn.net/u010859707/article/details/_blank)([alpha, binarize, ...]) | 朴素贝叶斯分类器多变量伯努利模型 |

 naive\_bayes.BernoulliNB（朴素贝叶斯分类器多变量伯努利模型），像MultinomialNB一样，这个分类是适用于离散数据。不同的是，在multinomialnb作品出现计数，BernoulliNB是专为二进制/布尔特征。naive\_bayes.MultinomialNB（朴素贝叶斯分类器多项式模型），多项式朴素贝叶斯分类器适用于具有离散特征的分类（例如，单词计数和文本分类）。多项式分布通常需要整数特征计数，然而，在实践中，分数计数，如国防军也可以工作。

## sklearn.neighbors: Nearest Neighbors（最近邻）

该sklearn.neighbors模块实现了k-最近邻居算法。

|  |  |
| --- | --- |
| [neighbors.NearestNeighbors](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html" \l "sklearn.neighbors.NearestNeighbors" \o "sklearn.neighbors.NearestNeighbors" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_neighbors, ...]) | 无监督学习者实施邻居搜索 |
| [neighbors.KNeighborsClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html" \l "sklearn.neighbors.KNeighborsClassifier" \o "sklearn.neighbors.KNeighborsClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 执行k-最近邻居的分类器投票 |
| [neighbors.RadiusNeighborsClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsClassifier.html" \l "sklearn.neighbors.RadiusNeighborsClassifier" \o "sklearn.neighbors.RadiusNeighborsClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 分类器在给定半径内的邻居中执行投票 |
| [neighbors.KNeighborsRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html" \l "sklearn.neighbors.KNeighborsRegressor" \o "sklearn.neighbors.KNeighborsRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_neighbors, ...]) | 基于k最近邻的回归 |
| [neighbors.RadiusNeighborsRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsRegressor.html" \l "sklearn.neighbors.RadiusNeighborsRegressor" \o "sklearn.neighbors.RadiusNeighborsRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([radius, ...]) | 基于固定半径内的邻居的回归 |
| [neighbors.NearestCentroid](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html" \l "sklearn.neighbors.NearestCentroid" \o "sklearn.neighbors.NearestCentroid" \t "http://blog.csdn.net/u010859707/article/details/_blank)([metric, ...]) | 最近质心分类器 |
| [neighbors.BallTree](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.BallTree.html" \l "sklearn.neighbors.BallTree" \o "sklearn.neighbors.BallTree" \t "http://blog.csdn.net/u010859707/article/details/_blank) | BallTree用于快速泛化N点问题 |
| [neighbors.KDTree](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html" \l "sklearn.neighbors.KDTree" \o "sklearn.neighbors.KDTree" \t "http://blog.csdn.net/u010859707/article/details/_blank) | KDTree用于快速泛化的N点问题 |
| [neighbors.LSHForest](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LSHForest.html" \l "sklearn.neighbors.LSHForest" \o "sklearn.neighbors.LSHForest" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_estimators, radius, ...]) | 使用LSH森林执行近似最近邻搜索 |
| [neighbors.DistanceMetric](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html" \l "sklearn.neighbors.DistanceMetric" \o "sklearn.neighbors.DistanceMetric" \t "http://blog.csdn.net/u010859707/article/details/_blank) | DistanceMetric类 |
| [neighbors.KernelDensity](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KernelDensity.html" \l "sklearn.neighbors.KernelDensity" \o "sklearn.neighbors.KernelDensity" \t "http://blog.csdn.net/u010859707/article/details/_blank)([bandwidth, ...]) | 核密度估计 |

neighbors.DistanceMetric（距离度量类），这个类提供了快速距离度量函数的统一接口，各指标可通过get\_metric类方法和度量字符串标识符。neighbors.KNeighborsRegressor（基于k最近邻的回归），通过对训练集中最近邻关联的目标进行局部插值来预测目标。neighbors.RadiusNeighborsRegressor（基于固定半径内的邻居的回归），目标是通过局部插值预测的目标相关的训练中的最近邻居集。neighbors.NearestCentroid（最重心分类器），每一个类由它的质心表示，测试样本用最近的质心来分类。

函数

|  |  |
| --- | --- |
| [neighbors.kneighbors\_graph](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.kneighbors_graph.html" \l "sklearn.neighbors.kneighbors_graph" \o "sklearn.neighbors.kneighbors_graph" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, n\_neighbors[, ...]) | 计算X中k个邻居的（加权）图 |
| [neighbors.radius\_neighbors\_graph](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.radius_neighbors_graph.html" \l "sklearn.neighbors.radius_neighbors_graph" \o "sklearn.neighbors.radius_neighbors_graph" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, radius) | 计算X中的点的邻居的（加权）图 |

## sklearn.neural\_network: Neural network models（神经网络模型）

该sklearn.neural\_network模块包括基于神经网络的模型。

|  |  |
| --- | --- |
| [neural\_network.BernoulliRBM](http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.BernoulliRBM.html" \l "sklearn.neural_network.BernoulliRBM" \o "sklearn.neural_network.BernoulliRBM" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_components, ...]) | 伯努利限制玻尔兹曼机（RBM） |
| [neural\_network.MLPClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html" \l "sklearn.neural_network.MLPClassifier" \o "sklearn.neural_network.MLPClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 多层感知器分类器 |
| [neural\_network.MLPRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html" \l "sklearn.neural_network.MLPRegressor" \o "sklearn.neural_network.MLPRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 多层感知器回归 |

  neural\_network.BernoulliRBM（伯努利限制玻尔兹曼机（RBM）），具有二元可见单位和二元隐藏单元的受限玻尔兹曼机。参数估计使用随机的最大似然（SML），也被称为持久性对比散度（PCD）[ 2 ]。neural\_network.MLPClassifier（多层感知器分类器），该模型优化日志损失函数采用LBFGS或随机梯度下降的方法。neural\_network.MLPRegressor（多层感知器的回归），该模型优化平方损失采用LBFGS或随机梯度下降。

## sklearn.pipeline: Pipeline（管道）

该sklearn.pipeline模块实现实用程序来构建复合估计器，作为变换链和估计器链。

|  |  |
| --- | --- |
| [pipeline.Pipeline](http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html" \l "sklearn.pipeline.Pipeline" \o "sklearn.pipeline.Pipeline" \t "http://blog.csdn.net/u010859707/article/details/_blank)(steps) | 最终估计量的变换管道 |
| [pipeline.FeatureUnion](http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.FeatureUnion.html" \l "sklearn.pipeline.FeatureUnion" \o "sklearn.pipeline.FeatureUnion" \t "http://blog.csdn.net/u010859707/article/details/_blank)(transformer\_list[, ...]) | 连接多个变压器对象的结果 |
| [pipeline.make\_pipeline](http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.make_pipeline.html" \l "sklearn.pipeline.make_pipeline" \o "sklearn.pipeline.make_pipeline" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*steps) | 从给定的估计量构建管道 |
| [pipeline.make\_union](http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.make_union.html" \l "sklearn.pipeline.make_union" \o "sklearn.pipeline.make_union" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*transformers) | 从给定的变压器构造一个FeatureUnion |

 pipeline.FeatureUnion（连接多个变压器对象的结果），这个估计采用并联变压器的对象列表中输入数据，然后将结果联系起来。这是对结合多种特征提取机制成一个单一的变压器是很有用的。变压器的参数可以设置使用其名称和参数名称由“\_\_分离”。同时也可以通过将参数的名称设置为另一个变压器或将其设置为非删除来完全替换变压器。

Pipeline.Pipeline（最终估计量的变换管道），依次应用转换列表和最终估计量。管道的中间步骤必须是反式的—种形式，也就是说，它们必须实现匹配和转换方法。最后的估计只需要实现拟合。管道中的变压器可以使用内存参数进行缓存。流水线的目的是在设置不同时组装几个可以交叉验证的步骤—耳鼻喉参数。为此，可以设置使用他们的名字和参数名称由“\_\_分隔的各个步骤的参数。

pipeline.make\_pipeline（从给定的估计量构建管道），这是管道构造器的简写；它不要求，也不允许命名估计量。相反，它们的名称将自动设置为小写类型。

pipeline.make\_union（从给定的变压器构造featureunion），这是一个为featureunion构造函数速记：它不需要，而不允许，命名变压器。相反，它们将根据它们的类型自动命名，它也不允许加权。

## sklearn.preprocessing: Preprocessing and Normalization（预处理和规范化）

该sklearn.preprocessing模块包括缩放，定心，归一化，二值化和插补方法。

|  |  |
| --- | --- |
| [preprocessing.Binarizer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Binarizer.html" \l "sklearn.preprocessing.Binarizer" \o "sklearn.preprocessing.Binarizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([threshold, copy]) | 根据阈值对数据进行二值化（将特征值设置为0或1） |
| [preprocessing.FunctionTransformer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.FunctionTransformer.html" \l "sklearn.preprocessing.FunctionTransformer" \o "sklearn.preprocessing.FunctionTransformer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([func, ...]) | 从任意可调用的构造一个变压器 |
| [preprocessing.Imputer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html" \l "sklearn.preprocessing.Imputer" \o "sklearn.preprocessing.Imputer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([missing\_values, ...]) | 用于完成缺失值的插补变压器 |
| [preprocessing.KernelCenterer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KernelCenterer.html" \l "sklearn.preprocessing.KernelCenterer" \o "sklearn.preprocessing.KernelCenterer" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 中心一个内核矩阵 |
| [preprocessing.LabelBinarizer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelBinarizer.html" \l "sklearn.preprocessing.LabelBinarizer" \o "sklearn.preprocessing.LabelBinarizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([neg\_label, ...]) | 以一对一的方式对标签进行二值化 |
| [preprocessing.LabelEncoder](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html" \l "sklearn.preprocessing.LabelEncoder" \o "sklearn.preprocessing.LabelEncoder" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 在0和n\_classes-1之间编码标签 |
| [preprocessing.MultiLabelBinarizer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MultiLabelBinarizer.html" \l "sklearn.preprocessing.MultiLabelBinarizer" \o "sklearn.preprocessing.MultiLabelBinarizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([classes, ...]) | 迭代的可迭代对象和细粒度的格式之间的转换 |
| [preprocessing.MaxAbsScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html" \l "sklearn.preprocessing.MaxAbsScaler" \o "sklearn.preprocessing.MaxAbsScaler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([copy]) | 按每个特征的最大绝对值进行缩放 |
| [preprocessing.MinMaxScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html" \l "sklearn.preprocessing.MinMaxScaler" \o "sklearn.preprocessing.MinMaxScaler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([feature\_range, copy]) | 通过将每个功能缩放到给定范围来转换功能 |
| [preprocessing.Normalizer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normalizer.html" \l "sklearn.preprocessing.Normalizer" \o "sklearn.preprocessing.Normalizer" \t "http://blog.csdn.net/u010859707/article/details/_blank)([norm, copy]) | 将样品归一化为单位范数 |
| [preprocessing.OneHotEncoder](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html" \l "sklearn.preprocessing.OneHotEncoder" \o "sklearn.preprocessing.OneHotEncoder" \t "http://blog.csdn.net/u010859707/article/details/_blank)([n\_values, ...]) | 使用一个单一的一个K方案来编码分类整数特征 |
| [preprocessing.PolynomialFeatures](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html" \l "sklearn.preprocessing.PolynomialFeatures" \o "sklearn.preprocessing.PolynomialFeatures" \t "http://blog.csdn.net/u010859707/article/details/_blank)([degree, ...]) | 生成多项式和交互特征 |
| [preprocessing.RobustScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html" \l "sklearn.preprocessing.RobustScaler" \o "sklearn.preprocessing.RobustScaler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([with\_centering, ...]) | 使用对异常值可靠的统计信息来缩放特征 |
| [preprocessing.StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html" \l "sklearn.preprocessing.StandardScaler" \o "sklearn.preprocessing.StandardScaler" \t "http://blog.csdn.net/u010859707/article/details/_blank)([copy, ...]) | 通过删除平均值和缩放到单位方差来标准化特征 |

preprocessing.Binarizer9（根据阈值对数据进行二值化（将特征值设置为0或1）），大于阈值映射的值为1，而小于或等于阈值映射的值为0。与默认阈值为0，只有正值映射为1。

二值化是对文本计数数据的一种常见操作，在这种情况下，分析人员可以决定只考虑某个特性的存在或不存在，而不是只考虑一个量化的出现次数。

它也可以用作考虑布尔随机变量的估计的预处理步骤（例如在贝叶斯环境中使用伯努利分布）。

preprocessing.FunctionTransformer（从任意可调用的构造一个变压器），一个functiontransformer提出X（或者Y）参数的用户定义函数或函数对象并返回这个函数的结果。这对于无状态转换如以频率的记录和做自定义缩放是有用的。且一个functiontransformer不会做任何检查其功能的输出。

preprocessing.KernelCenterer（中心一个内核矩阵），设k（x，z）是φ（x）定义的核，φ是希尔伯特空间的函数映射x。kernelcenterer中心（即规范化具有零均值）没有明确计算φ（x）的数据。它是相当于定心φ（x）与sklearn的预处理。

preprocessing.LabelBinarizer（以一对一的方式对标签进行二值化），几个回归和二分类算法在scikit可用。将这些算法扩展到多类分类的一个简单方法是使用所谓的“一对一”方案。在学习的时候，这仅仅是在包括学习一个自变量或二进制分类器。这样做时，需要将多类标签转换为二进制标签（属于或不属于类）。labelbinarizer用变换方法使这个过程变得简单。

preprocessing.MultiLabelBinarizer（迭代的可迭代对象和细粒度的格式之间的转换），虽然一个列表或元组细粒度的数据集是一个非常直观的格式，但是这在编程的时候非常不方便。该变压器将这种直观的格式，支持细粒度的格式之间：一个（样本x班）的二进制矩阵表示一类标签的存在。

preprocessing.MaxAbsScaler（按每个特征的最大绝对值进行缩放），这个估计尺度转化为每一个单独的功能，在训练集的每个特征的最大绝对值为1。它不转移中心的数据，从而不破坏任何稀疏。这个计数器也可以应用于稀疏矩阵的CSR或CSC。

preprocessing.MinMaxScaler（通过将每个特征缩放到给定范围来变换特征），该估计量对每个特征进行缩放和平移，使其在训练集的给定范围内，即零和一。

preprocessing.Normalizer（将样品归一化为单位范数），每个样品（即每行数据矩阵）至少有一个非零分量法独立于其他的样品，其规范（L1或L2）等于一。这种变压器是能够工作在密集NumPy数组和scipy.sparse矩阵（如果你想避免复制/转换的负担使用CSR格式）。

preprocessing.PolynomialFeatures（生成多项式和交互特征），生成一个新的特征矩阵的特征多项式组成的所有组合的程度小于或等于指定的程度。例如，如果输入样本是二维的，并且形式为[ A，B ]，则2阶多项式特征[ 1，A，B，一个^ 2，AB，B ^ 2 ]。

preprocessing.StandardScaler（通过除去均值和单位方差来标准化特征），通过计算样本的相关统计数据，独立地对每个特征进行中心化和缩放。在训练集中。然后用转换方法将均值和标准偏差存储在以后的数据上。数据集的标准化是许多机器学习估计的共同要求：如果个体特征不像标准的正态分布数据（如高斯均值为0均值和单位方差），它们可能表现得很差。

函数

|  |  |
| --- | --- |
| [preprocessing.add\_dummy\_feature](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.add_dummy_feature.html" \l "sklearn.preprocessing.add_dummy_feature" \o "sklearn.preprocessing.add_dummy_feature" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, value]) | 增强数据集，带有额外的虚拟功能 |
| [preprocessing.binarize](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.binarize.html" \l "sklearn.preprocessing.binarize" \o "sklearn.preprocessing.binarize" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, threshold, copy]) | 数组式或scipy.sparse矩阵的布尔阈值 |
| [preprocessing.label\_binarize](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.label_binarize.html" \l "sklearn.preprocessing.label_binarize" \o "sklearn.preprocessing.label_binarize" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y, classes[, ...]) | 以 one-vs-all 的方式对标签进行二值化 |
| [preprocessing.maxabs\_scale](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.maxabs_scale.html" \l "sklearn.preprocessing.maxabs_scale" \o "sklearn.preprocessing.maxabs_scale" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, axis, copy]) | 将每个特征缩放到[-1,1]范围，而不破坏稀疏度 |
| [preprocessing.minmax\_scale](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.minmax_scale.html" \l "sklearn.preprocessing.minmax_scale" \o "sklearn.preprocessing.minmax_scale" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, ...]) | 通过将每个功能缩放到给定范围来转换功能 |
| [preprocessing.normalize](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html" \l "sklearn.preprocessing.normalize" \o "sklearn.preprocessing.normalize" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, norm, axis, ...]) | 将输入向量分别缩放到单位范数（向量长度） |
| [preprocessing.robust\_scale](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.robust_scale.html" \l "sklearn.preprocessing.robust_scale" \o "sklearn.preprocessing.robust_scale" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, axis, ...]) | 沿着任何轴标准化数据集 |
| [preprocessing.scale](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html" \l "sklearn.preprocessing.scale" \o "sklearn.preprocessing.scale" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X[, axis, with\_mean, ...]) | 沿着任何轴标准化数据集 |

## sklearn.random\_projection: Random projection（随机投影）

随机投影是一种简单且计算有效的方法，通过交易控制的精确度（作为附加方差）来减少数据的维度，以实现更快的处理时间和更小的模型大小。

控制随机投影矩阵的维数和分布，以保留数据集的任意两个样本之间的成对距离。随机投影效率背后的主要理论结果是 Johnson-Lindenstrauss lemma（引用维基百科）：

在数学方面，Johnson-Lindenstrauss引理是从高维度到低维度欧几里德空间的低失真嵌入点的结果。引理指出，高维度空间中的一小部分点可以嵌入到较低维度的空间中，使得点之间的距离几乎保持不变。用于嵌入的地图至少为Lipschitz，甚至可以被视为正交投影。

|  |  |
| --- | --- |
| [random\_projection.GaussianRandomProjection](http://scikit-learn.org/stable/modules/generated/sklearn.random_projection.GaussianRandomProjection.html" \l "sklearn.random_projection.GaussianRandomProjection" \o "sklearn.random_projection.GaussianRandomProjection" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 通过高斯随机投影降低维数 |
| [random\_projection.SparseRandomProjection](http://scikit-learn.org/stable/modules/generated/sklearn.random_projection.SparseRandomProjection.html" \l "sklearn.random_projection.SparseRandomProjection" \o "sklearn.random_projection.SparseRandomProjection" \t "http://blog.csdn.net/u010859707/article/details/_blank)([...]) | 通过稀疏随机投影降低维数 |
| [random\_projection.johnson\_lindenstrauss\_min\_dim](http://scikit-learn.org/stable/modules/generated/sklearn.random_projection.johnson_lindenstrauss_min_dim.html" \l "sklearn.random_projection.johnson_lindenstrauss_min_dim" \o "sklearn.random_projection.johnson_lindenstrauss_min_dim" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 找到一个“安全”数量的组件随机投影到 |

  random\_projection.GaussianRandomProjection（通过高斯随机投影降低维数），随机矩阵的成分都来自N（0, 1 / n\_components）。

random\_projection.SparseRandomProjection（通过稀疏随机投影降低维数），稀疏随机矩阵是一种替代密集随机投影矩阵，保证了类似的嵌入质量，同时具有更高的内存效率，并允许更快的计算投影数据。

random\_projection.johnson\_lindenstrauss\_min\_dim（找到一个“安全”数量的组件随机投影到），注意，维度的数量与原始的特征数无关，而取决于数据集的大小：数据集越大，EPS嵌入的最小维数越高。

函数

|  |  |
| --- | --- |
| fit(X[, y]) | 生成稀疏随机投影矩阵 |
| fit\_transform(X[, y]) | 找到适合的数据，然后转换它 |
| get\_params([deep]) | 获取此估计器的参数 |
| set\_params(\*\*params) | 设置该估计器的参数 |
| transform(X) | 用矩阵乘积与随机矩阵求出数据 |

## sklearn.semi\_supervised Semi-Supervised Learning（半监督学习）

该sklearn.semi\_supervised模块实现半监督学习算法。这些算法使用少量的标记数据和大量未标记的分类任务数据。该模块包括标签传播。

|  |  |
| --- | --- |
| [semi\_supervised.LabelPropagation](http://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelPropagation.html" \l "sklearn.semi_supervised.LabelPropagation" \o "sklearn.semi_supervised.LabelPropagation" \t "http://blog.csdn.net/u010859707/article/details/_blank)([kernel, ...]) | 标签传播分类器 |
| [semi\_supervised.LabelSpreading](http://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelSpreading.html" \l "sklearn.semi_supervised.LabelSpreading" \o "sklearn.semi_supervised.LabelSpreading" \t "http://blog.csdn.net/u010859707/article/details/_blank)([kernel, ...]) | 用于半监督学习的LabelSpread模型 |

## sklearn.svm: Support Vector Machines（支持向量机）

该sklearn.svm模块包括支持向量机算法

|  |  |
| --- | --- |
| [svm.SVC](http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html" \l "sklearn.svm.SVC" \o "sklearn.svm.SVC" \t "http://blog.csdn.net/u010859707/article/details/_blank)([C, kernel, degree, gamma, coef0, ...]) | 支持向量分类 |
| [svm.LinearSVC](http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html" \l "sklearn.svm.LinearSVC" \o "sklearn.svm.LinearSVC" \t "http://blog.csdn.net/u010859707/article/details/_blank)([penalty, loss, dual, tol, C, ...]) | 线性支持向量分类 |
| [svm.NuSVC](http://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVC.html" \l "sklearn.svm.NuSVC" \o "sklearn.svm.NuSVC" \t "http://blog.csdn.net/u010859707/article/details/_blank)([nu, kernel, degree, gamma, ...]) | Nu支持向量分类 |
| [svm.SVR](http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html" \l "sklearn.svm.SVR" \o "sklearn.svm.SVR" \t "http://blog.csdn.net/u010859707/article/details/_blank)([kernel, degree, gamma, coef0, tol, ...]) | ε支持向量回归 |
| [svm.LinearSVR](http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html" \l "sklearn.svm.LinearSVR" \o "sklearn.svm.LinearSVR" \t "http://blog.csdn.net/u010859707/article/details/_blank)([epsilon, tol, C, loss, ...]) | 线性支持向量回归 |
| [svm.NuSVR](http://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVR.html" \l "sklearn.svm.NuSVR" \o "sklearn.svm.NuSVR" \t "http://blog.csdn.net/u010859707/article/details/_blank)([nu, C, kernel, degree, gamma, ...]) | Nu支持向量回归 |
| [svm.OneClassSVM](http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html" \l "sklearn.svm.OneClassSVM" \o "sklearn.svm.OneClassSVM" \t "http://blog.csdn.net/u010859707/article/details/_blank)([kernel, degree, gamma, ...]) | 无监督异常检测 |
| [svm.l1\_min\_c](http://scikit-learn.org/stable/modules/generated/sklearn.svm.l1_min_c.html" \l "sklearn.svm.l1_min_c" \o "sklearn.svm.l1_min_c" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y[, loss, fit\_intercept, ...]) | 返回C的最低边界，使得对于C（l1\_min\_C，无穷大），模型保证不为空 |

svm.LinearSVC(线性支持向量分类)，类似于SVC参数= 'linear '，但实现从liblinear而不是LIBSVM，因此在处罚和损失函数的选择上更具灵活性同时应该选择规模应该更大数量的样本。这个类支持密集和稀疏输入，并且多类支持是根据一个与另一个方案一起处理的。

svm.LinearSVR（线性支持向量回归），类似于SVR参数= 'linear '，但实现从liblinear而不是LIBSVM，因此在处罚和损失函数的选择上更具灵活性同时应该选择规模应该更大数量的样本。这个类支持密集和稀疏输入。

svm.NuSVC（Nu支持向量分类），与SVC类似，但使用参数来控制支持向量的数量。这种方法的实现是基于LIBSVM。

svm.NuSVR（Nu支持向量回归），对于回归来说，其类似于NuSVC，使用一个参数NU控制支持向量的数目。然而，不像nusvc，其中Nu取代C，这里的NU取代的是SVR参数ε。这种方法的实现是基于LIBSVM。

svm.OneClassSVM（无监督异常检测），支持估计高维分布，这种方法的实现是基于LIBSVM。

svm.SVC（支持向量分类），这个类的实现是基于LIBSVM。随着样本数的增加，拟合时间复杂度超过二次，这使得样本数超过10000个样本很难扩展到数据集。

svm.SVR（ε支持向量回归），模型中的自由参数为c和ε。这个类的实现是基于LIBSVM。

Low-level methods（低级方法）

|  |  |
| --- | --- |
| [svm.libsvm.fit](http://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.fit.html" \l "sklearn.svm.libsvm.fit" \o "sklearn.svm.libsvm.fit" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 使用libsvm（低级方法）训练模型 |
| [svm.libsvm.decision\_function](http://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.decision_function.html" \l "sklearn.svm.libsvm.decision_function" \o "sklearn.svm.libsvm.decision_function" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 预测利润（这是libsvm的名称是predict\_values） |
| [svm.libsvm.predict](http://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.predict.html" \l "sklearn.svm.libsvm.predict" \o "sklearn.svm.libsvm.predict" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 给定模型预测X的目标值（低级方法） |
| [svm.libsvm.predict\_proba](http://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.predict_proba.html" \l "sklearn.svm.libsvm.predict_proba" \o "sklearn.svm.libsvm.predict_proba" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 预测概率 |
| [svm.libsvm.cross\_validation](http://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.cross_validation.html" \l "sklearn.svm.libsvm.cross_validation" \o "sklearn.svm.libsvm.cross_validation" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 交叉验证程序的绑定（低级程序） |

  svm.libsvm.decision\_function（预测利润）我们必须重建模型和参数，以确保我们与Python对象保持同步。

svm.libsvm.predict\_proba（预测概率），svm\_model存储了来预测一个给定值的所有参数。同样的，我们必须重建模型和参数，以确保我们与Python对象保持同步。

## sklearn.tree: Decision Trees（决策树）

该sklearn.tree模块包括用于分类和回归的基于决策树的模型。

|  |  |
| --- | --- |
| [tree.DecisionTreeClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html" \l "sklearn.tree.DecisionTreeClassifier" \o "sklearn.tree.DecisionTreeClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([criterion, ...]) | 决策树分类器 |
| [tree.DecisionTreeRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html" \l "sklearn.tree.DecisionTreeRegressor" \o "sklearn.tree.DecisionTreeRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([criterion, ...]) | 决策树倒数 |
| [tree.ExtraTreeClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeClassifier.html" \l "sklearn.tree.ExtraTreeClassifier" \o "sklearn.tree.ExtraTreeClassifier" \t "http://blog.csdn.net/u010859707/article/details/_blank)([criterion, ...]) | 一个非常随机的树分类器 |
| [tree.ExtraTreeRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeRegressor.html" \l "sklearn.tree.ExtraTreeRegressor" \o "sklearn.tree.ExtraTreeRegressor" \t "http://blog.csdn.net/u010859707/article/details/_blank)([criterion, ...]) | 一个非常随机的树回归 |
| [tree.export\_graphviz](http://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html" \l "sklearn.tree.export_graphviz" \o "sklearn.tree.export_graphviz" \t "http://blog.csdn.net/u010859707/article/details/_blank) | 以DOT格式导出决策树 |

 tree.ExtraTreeClassifier（一个非常随机的树分类器），Extra-trees不同于经典决策树的构建方式，当寻找到单独的一个节点去分割一个样本为两组时，随机将绘制的每个max\_features和随机选定的功能，选择最佳的分割。当max\_features设置为1，这相当于建立一个完全随机决策树。

tree.ExtraTreeRegressor（一个非常随机的树回归），Extra-trees不同于经典决策树的构建方式，当寻找到单独的一个节点去分割一个样本为两组时，随机将绘制的每个max\_features和随机选定的功能，选择最佳的分割。当max\_features设置为1，这相当于建立一个完全随机决策树。

tree.export\_graphviz（以DOT格式导出决策树），这个函数生成的决策树的图像表示，并写进out\_file。一旦导出，可以使用图形生成图。

函数

|  |  |
| --- | --- |
| apply(X[, check\_input]) | 返回的叶，每个样本被预测为指标 |
| decision\_path(X[, check\_input]) | 返回树中的决策路径 |
| fit(X, y[, sample\_weight, check\_input, ...]) | 建立一个决策树分类器的训练集（x，y） |
| get\_params([deep]) | 获取此估计器的参数 |
| predict(X[, check\_input]) | 为x预测类或回归值 |
| predict\_log\_proba(X) | 预测输入样本x的对数的类概率 |
| predict\_proba(X[, check\_input]) | 预测输入样本x的类概率 |
| score(X, y[, sample\_weight]) | 返回给定测试数据和标签的平均精度 |
| set\_params(\*\*params) | 设置该估计器的参数 |

## sklearn.utils: Utilities（工具）

该sklearn.utils模块包括各种实用程序。

|  |  |
| --- | --- |
| utils.as\_float\_array(X[, copy, force\_all\_finite]) | 将数组转换为浮动数组 |
| utils.assert\_all\_finite(X) | 如果X包含NaN或infinity,就抛出ValueErro |
| utils.check\_X\_y(X, y[, accept\_sparse, ...]) | 标准估计量的输入验证 |
| utils.check\_array(array[, accept\_sparse, ...]) | 数组、列表、稀疏矩阵或类似的输入验证 |
| utils.check\_consistent\_length(\*arrays) | 检查所有数组是否具有一致的第一个维度 |
| utils.check\_random\_state(seed) | 把种子变成一np.random.randomstate实例 |
| utils.class\_weight.  compute\_class\_weight(...) | 非平衡数据集的类权重估计 |
| utils.class\_weight.  compute\_sample\_weight(...) | 不平衡数据集按类估计样本权重 |
| utils.estimator\_checks.  check\_estimator(Estimator) | 检查估计坚持scikit学习公约 |
| utils.extmath.safe\_sparse\_dot(a, b[, ...]) | 正确处理稀疏矩阵情况的点乘积 |
| utils.indexable(\*iterables) | 使阵列可用于交叉验证 |
| utils.resample(\*arrays, \*\*options) | 以一致的方式重采样数组或稀疏矩阵 |
| utils.safe\_indexing(X, indices) | 使用索引返回x中的项或行 |
| utils.shuffle(\*arrays, \*\*options) | 以一致的方式洗牌数组或稀疏矩阵 |
| utils.sparsefuncs.  incr\_mean\_variance\_axis(X, ...) | 计算增量的均值和方差，沿着一个轴在CSR或CSC矩阵 |
| utils.sparsefuncs.  inplace\_column\_scale(X, scale) | 一柱一CSC /企业社会责任矩阵缩放 |
| utils.sparsefuncs.inplace\_row\_scale(X,  scale) | 某行缩放的CSR或CSC矩阵 |
| utils.sparsefuncs.inplace\_swap\_row(X, m,  n) | 交换两行的一个CSC / CSR矩阵 |
| utils.sparsefuncs.  inplace\_swap\_column(X, m, n) | 交换一个CSC / CSR矩阵的两个列 |
| utils.sparsefuncs.mean\_variance\_axis(X,axis) | 计算均值和方差，沿着一个轴在CSR或CSC矩阵 |
| utils.validation.check\_is\_fitted(estimator,  ...) | 估计执行is\_fitted验证 |
| utils.validation.check\_symmetric(array[,  ...]) | 确保数组是二维，平方和对称 |
| utils.validation.column\_or\_1d(y[, warn]) | Ravel柱或1D NumPy数组，否则将引发错误 |
| utils.validation.has\_fit\_parameter(...) | 检查估计器的拟合方法是否支持给定参数。 |

utils.as\_float\_array（将数组转换为浮动数组），新的D类型将np.float32或np.float64，取决于原始类型。函数可以根据参数复制创建一个副本或修改参数。

utils.check\_array（数组、列表、稀疏矩阵或类似的输入验证），默认情况下，输入转换为至少二维NumPy数组。如果数组中的D为对象，尝试将上浮，提高故障。

utils.estimator\_checks.check\_estimator（）该估计器将运行一个广泛的测试套件，用于输入验证、形状等分类器的附加测试，回归、聚类或变压器，将运行如果估计类从相应的klearn.base继承。此测试可应用于类或实例。类当前有一些与构建相关的附加测试，而传递实例则允许测试多个选项。

utils.indexable（使阵列可用于交叉验证），检查一致长度，不通过任何一个，并确保一切都可以通过转换稀疏来索引。矩阵的CSR和转换非可积的对象数组。

utils.sparsefuncs.inplace\_column\_scale（一柱一CSC /企业社会责任矩阵缩放），通过调用调用者提供的特定尺度相乘数据矩阵的每一个特征，（n\_samples，n\_features）形状。

utils.validation.check\_is\_fitted（估计执行is\_fitted验证）检查估计拟合验证是否存在“all\_or\_any”传递的属性，提出了一个与给定的消息notfittederror。

## Recently deprecated（最近弃用）

To be removed in 0.21

|  |  |
| --- | --- |
| linear\_model.RandomizedLasso(\*args,  \*\*kwargs) | 随机的套索 |
| linear\_model.RandomizedLogisticRegression(...) | 随机Logistic回归 |
| neighbors.LSHForest([n\_estimators, radius, ...]) | 利用LSH森林执行近似最近邻搜索 |

linear\_model.RandomizedLasso（随机的套索），随机抽样的套索的训练数据，计算一个套索估计系数的随机子集的惩罚被缩放。通过执行这个双随机几次，方法分配分数较高，多次入选在随机化的特点。这就是所谓的稳定性选择。简而言之，更多选择的特征被认为是好的特性。

neighbors.LSHForest（利用LSH森林执行近似最近邻搜索），LSH森林：是香草近似最近的一种替代方法邻域搜索法。LSH森林数据结构已使用已排序数组和二进制的实现

搜索和32位固定长度的哈希值。

To be removed in 0.2

|  |  |
| --- | --- |
| [grid\_search.ParameterGrid](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.ParameterGrid.html" \l "sklearn.grid_search.ParameterGrid" \o "sklearn.grid_search.ParameterGrid" \t "http://blog.csdn.net/u010859707/article/details/_blank)(param\_grid) | 每个参数的网格具有离散数量的值 |
| [grid\_search.ParameterSampler](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.ParameterSampler.html" \l "sklearn.grid_search.ParameterSampler" \o "sklearn.grid_search.ParameterSampler" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...[, random\_state]) | 发电机对从给定分布采样的参数 |
| [grid\_search.GridSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html" \l "sklearn.grid_search.GridSearchCV" \o "sklearn.grid_search.GridSearchCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, param\_grid) | 对估计器的指定参数值进行详尽搜索 |
| [grid\_search.RandomizedSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.RandomizedSearchCV.html" \l "sklearn.grid_search.RandomizedSearchCV" \o "sklearn.grid_search.RandomizedSearchCV" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, ...) | 随机搜索超参数 |
| [cross\_validation.LeaveOneOut](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.LeaveOneOut.html" \l "sklearn.cross_validation.LeaveOneOut" \o "sklearn.cross_validation.LeaveOneOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n) |  |
| [cross\_validation.LeavePOut](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.LeavePOut.html" \l "sklearn.cross_validation.LeavePOut" \o "sklearn.cross_validation.LeavePOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n, p) | Leave-P-Out交叉验证迭代器 |
| [cross\_validation.KFold](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.KFold.html" \l "sklearn.cross_validation.KFold" \o "sklearn.cross_validation.KFold" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n[, n\_folds, ...]) | K倍交叉验证迭代器 |
| [cross\_validation.LabelKFold](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.LabelKFold.html" \l "sklearn.cross_validation.LabelKFold" \o "sklearn.cross_validation.LabelKFold" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels[, n\_folds]) | 具有非重叠标签的K-fold迭代器变体 |
| [cross\_validation.LeaveOneLabelOut](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.LeaveOneLabelOut.html" \l "sklearn.cross_validation.LeaveOneLabelOut" \o "sklearn.cross_validation.LeaveOneLabelOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels) | Leave-One-Label\_Out交叉验证迭代器 |
| [cross\_validation.LeavePLabelOut](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.LeavePLabelOut.html" \l "sklearn.cross_validation.LeavePLabelOut" \o "sklearn.cross_validation.LeavePLabelOut" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels, p) | Leave-P-Label\_Out交叉验证迭代器 |
| [cross\_validation.LabelShuffleSplit](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.LabelShuffleSplit.html" \l "sklearn.cross_validation.LabelShuffleSplit" \o "sklearn.cross_validation.LabelShuffleSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)(labels[, ...]) | Shuffle-Labels-Out交叉验证迭代器 |
| [cross\_validation.StratifiedKFold](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedKFold.html" \l "sklearn.cross_validation.StratifiedKFold" \o "sklearn.cross_validation.StratifiedKFold" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y[, ...]) | 分层K-折叠交叉验证迭代器 |
| [cross\_validation.ShuffleSplit](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.ShuffleSplit.html" \l "sklearn.cross_validation.ShuffleSplit" \o "sklearn.cross_validation.ShuffleSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)(n[, n\_iter, ...]) | 随机置换交叉验证迭代器 |
| [cross\_validation.StratifiedShuffleSplit](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html" \l "sklearn.cross_validation.StratifiedShuffleSplit" \o "sklearn.cross_validation.StratifiedShuffleSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)(y[, ...]) | 分层ShuffleSplit交叉验证迭代器 |
| [cross\_validation.PredefinedSplit](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.PredefinedSplit.html" \l "sklearn.cross_validation.PredefinedSplit" \o "sklearn.cross_validation.PredefinedSplit" \t "http://blog.csdn.net/u010859707/article/details/_blank)(test\_fold) | 预定义的分割交叉验证迭代器 |
| [decomposition.RandomizedPCA](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.RandomizedPCA.html" \l "sklearn.decomposition.RandomizedPCA" \o "sklearn.decomposition.RandomizedPCA" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*args, \\*\\*kwargs) | 主成分分析（PCA）使用随机SVD |
| [gaussian\_process.GaussianProcess](http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcess.html" \l "sklearn.gaussian_process.GaussianProcess" \o "sklearn.gaussian_process.GaussianProcess" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*args, \\*\\*kwargs) | 遗留高斯过程模型类 |
| [mixture.GMM](http://scikit-learn.org/stable/modules/generated/sklearn.mixture.GMM.html" \l "sklearn.mixture.GMM" \o "sklearn.mixture.GMM" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*args, \\*\\*kwargs) | 传统高斯混合模型 |
| [mixture.DPGMM](http://scikit-learn.org/stable/modules/generated/sklearn.mixture.DPGMM.html" \l "sklearn.mixture.DPGMM" \o "sklearn.mixture.DPGMM" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*args, \\*\\*kwargs) | Dirichlet过程高斯混合模型 |
| [mixture.VBGMM](http://scikit-learn.org/stable/modules/generated/sklearn.mixture.VBGMM.html" \l "sklearn.mixture.VBGMM" \o "sklearn.mixture.VBGMM" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*args, \\*\\*kwargs) | 高斯混合模型的变分推理 |

cross\_validation.KFold（K倍交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection.kfold来替代。提供列车测试数据分割数据。将数据集分割成k连续折叠（默认情况下无需重排），然后每个折叠被用作验证集，K - 1剩余折叠形成训练集。

cross\_validation.LabelKFold（k的迭代器的变异与非重叠的标签），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection.groupkfold来替代。相同的标签不会出现在两个不同的折叠中（不同标签的数量必须至少等于折叠的数量）。折叠是近似平衡的，在不同的折叠中，不同的标记的数量大致相同。

cross\_validation.LeaveOneLabelOut（Leave-One-Label\_Out交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection。Leaveonegroupout来替代。提供根据第三方提供的标签来分割数据的列车/测试指标。这个标签的信息可以用来编码任意特定领域层样品为整数。

cross\_validation.LeaveOneOut（一次性交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection leaveoneout来替代。提供列车测试数据分割数据。每个样本作为测试集（单件）使用一次，其余的样本形成训练集。

cross\_validation.LeavePOut（leave-p-out交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection leavepout来替代。提供列车测试数据分割数据。这就导致对所有不同大小的p样本进行测试，剩下的N - P样本在每次迭代中形成训练集。

cross\_validation.LeavePLabelOut（leave-p-label\_out交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection.LeavePGroupsOut来替代。

cross\_validation.LabelShuffleSplit（Shuffle-Labels-Out交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection.GroupShuffleSplit来替代。根据第三方提供组提供随机的火车/测试指标拆分数据，这组信息—信息可以用来对任意特定领域层样品为整数。

cross\_validation.StratifiedKFold（分层K-折叠交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection.stratifiedkfold来替代。提供列车测试数据分割数据。这种交叉验证的对象是一个变化K倍返回分层折叠。折叠是通过保持每类样品的百分比来制作的。

cross\_validation.StratifiedShuffleSplit（Stratified ShuffleSplit交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection StratifiedShuffleSplit代替。提供列车测试数据分割数据。这种交叉验证的对象是合并stratifiedkfold和shufflesplit，返回分层随机折叠。折叠是通过保持每类样品的百分比来制作的。

cross\_validation.PredefinedSplit（预定义的交叉交叉验证迭代器），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection PredefinedSplit代替。根据预定义的方案将数据分割成训练/测试集折叠。每个样品可以分配给一个测试集折，所指定的用户通过test\_fold参数。

decomposition.RandomizedPCA（主成分分析（PCA）使用随机SVD），推荐使用0.18以后的版本：这个类将被删除0.20。使用PCA参数svd\_solver '随机'代替。利用数据的近似奇异值分解，只保留最重要的奇异向量，将数据投影到较低维空间，从而减少线性维数。

gaussian\_process.GaussianProcess（遗留高斯过程模型类），推荐使用0.18以后的版本：这个类将被删除0.20。使用gaussianprocessregressor替代。

grid\_search.ParameterGrid（每个变量的离散值的参数网格），使用0.18以后的版本：该模块将被删除0.20。使用sklearn.model\_selection parametergrid替代。且可用于用Python内置函数ITER迭代参数值组合。

函数

|  |  |
| --- | --- |
| [grid\_search.fit\_grid\_point](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.fit_grid_point.html" \l "sklearn.grid_search.fit_grid_point" \o "sklearn.grid_search.fit_grid_point" \t "http://blog.csdn.net/u010859707/article/details/_blank)(X, y, estimator, ...) | 适合一组参数 |
| [learning\_curve.learning\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.learning_curve.learning_curve.html" \l "sklearn.learning_curve.learning_curve" \o "sklearn.learning_curve.learning_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, X, y) | 学习曲线 |
| [learning\_curve.validation\_curve](http://scikit-learn.org/stable/modules/generated/sklearn.learning_curve.validation_curve.html" \l "sklearn.learning_curve.validation_curve" \o "sklearn.learning_curve.validation_curve" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, ...) | 验证曲线 |
| [cross\_validation.cross\_val\_predict](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.cross_val_predict.html" \l "sklearn.cross_validation.cross_val_predict" \o "sklearn.cross_validation.cross_val_predict" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, X) | 为每个输入数据点生成交叉验证的估计 |
| [cross\_validation.cross\_val\_score](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.cross_val_score.html" \l "sklearn.cross_validation.cross_val_score" \o "sklearn.cross_validation.cross_val_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(estimator, X) | 通过交叉验证评估分数 |
| [cross\_validation.check\_cv](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.check_cv.html" \l "sklearn.cross_validation.check_cv" \o "sklearn.cross_validation.check_cv" \t "http://blog.csdn.net/u010859707/article/details/_blank)(cv[, X, y, classifier]) | 输入检查器实用程序以用户友好的方式构建简历 |
| [cross\_validation.permutation\_test\_score](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.permutation_test_score.html" \l "sklearn.cross_validation.permutation_test_score" \o "sklearn.cross_validation.permutation_test_score" \t "http://blog.csdn.net/u010859707/article/details/_blank)(...) | 评估具有置换的交叉验证分数的意义 |
| [cross\_validation.train\_test\_split](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.train_test_split.html" \l "sklearn.cross_validation.train_test_split" \o "sklearn.cross_validation.train_test_split" \t "http://blog.csdn.net/u010859707/article/details/_blank)(\\*arrays, ...) | 将阵列或矩阵拆分成随机列和测试子集 |