# 机器学习对不平衡数据学习处理方案

## 机器学习中不平衡数据集的工具箱

软件包：imbalanced-learn0.3.2

介绍文档：https://pypi.python.org/pypi/imbalanced-learn/

API文档：http://contrib.scikit-learn.org/imbalanced-learn/stable/

详细API：http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html

## 包介绍

imbalanced-learn是一个python软件包，

提供了一些数据集中常用的重新采样技术，显示出强烈的不平衡性。它与scikit-learn兼容，是scikit-learn-contrib 项目的一部分。

测试imbalanced-learn是在Python 2.7和Python 3.5以及3.6下工作的。依赖关系需求基于最后的scikit-learn版本：

SciPy的（> = 0.13.3）

numpy的（> = 1.8.2）

scikit学习（> = 0.19.0）

另外，为了运行这些例子，你需要matplotlib（> = 2.0.0）。

不平衡学习目前在PyPi的存储库上可用，你可以通过点安装它：

## 包安装

pip install -U imbalanced-learn

import imblearn as ll

该软件包也在Anaconda Cloud平台发布

**更新sklearn到0.19以上**

from imblearn.over\_sampling import SMOTE

I'm using Jupyter Notebooks but because I have Windows I cannot use the terminal that came with Jupyter. I have anaconda so on the anaconda prompt I wrote:

pip install imblearn

Afterwards, I upgraded sklearn (because I've got an error of that):

pip install --upgrade sklearn

The upgrade and the install ran well in Anaconda prompt.

## 需求重述

大多数分类算法只能在每个类的样本数大致相同的情况下才能达到最佳效果。高度倾斜的数据集，其中少数人数量严重超过一个或多个类，已被证明是一个挑战，同时变得越来越普遍。

解决这个问题的一种方法是对数据集进行重新采样，以抵消这种不平衡，希望达到一个更为稳健和公平的决策边界。

重新抽样技术分为两类：

1.对大多数class进行抽样采样。

2.过度抽样少数class类别。

3.结合过度采样和欠采样。

4.创建合奏平衡套。

以下列出了当前在这个模块中实现的方法。

### 过采样Over-sampling

随机少数过度抽样与替换

SMOTE - 合成少数过采样技术[8]

bSMOTE（1＆2） - 类型1和2的边界线SMOTE [9]

SVM SMOTE - 支持载体SMOTE [10]

ADASYN - 适应不平衡学习的自适应综合抽样方法[15]

过采样然后是欠采样Over-sampling followed by under-sampling

SMOTE + Tomek

SMOTE + ENN

Ensemble sampling合奏采样

EasyEnsemble

BalanceCascade

在sphinx-gallery中提供了不同的算法。

### 过采样参考代码：

**from sklearn.datasets import make\_classification  
X, y = make\_classification(n\_samples=5000, n\_features=2, n\_informative=2,  
 n\_redundant=0, n\_repeated=0, n\_classes=3,  
 n\_clusters\_per\_class=1,  
 weights=[0.01, 0.05, 0.94],  
 class\_sep=0.8, random\_state=0)  
from collections import Counter  
print(sorted(Counter(y).items()))  
from imblearn.over\_sampling import RandomOverSampler,SMOTE  
# ros = RandomOverSampler(random\_state=0)  
ros = SMOTE()  
X\_resampled, y\_resampled = ros.fit\_sample(X, y)  
print(sorted(Counter(y\_resampled).items()))  
# [(0, 4674), (1, 4674), (2, 4674)]  
  
from sklearn.svm import LinearSVC  
clf = LinearSVC()  
clf.fit(X\_resampled, y\_resampled) # doctest : +ELLIPSIS  
  
print(clf.score(X\_resampled, y\_resampled))**

### 欠采样Under-sampling

随机抽样替换多数

提取大多数少数Tomek链接[1]

采用群集质心进行欠采样

NearMiss-（1＆2＆3）[2]

Condensend Nearest Neighbor [3]

单面选择[4]

Neighboorhood清理规则[5]

编辑最近的邻居[6]

实例硬度阈值[7]

重复编辑最近的邻居[14]

AllKNN [14]

### 欠采样参考代码：

**from collections import Counter  
from sklearn.datasets import make\_classification  
X, y = make\_classification(n\_samples=5000, n\_features=2, n\_informative=2,  
 n\_redundant=0, n\_repeated=0, n\_classes=3,  
 n\_clusters\_per\_class=1,  
 weights=[0.01, 0.05, 0.94],  
 class\_sep=0.8, random\_state=0)  
print(sorted(Counter(y).items()))  
# [(0, 64), (1, 262), (2, 4674)]  
from imblearn.under\_sampling import ClusterCentroids  
cc = ClusterCentroids(random\_state=0)  
X\_resampled, y\_resampled = cc.fit\_sample(X, y)  
print(sorted(Counter(y\_resampled).items()))  
# [(0, 64), (1, 64), (2, 64)]**

## 参考文献：

[1] ：I. Tomek，“CNN的两种修改”，In Systems，Man，and Cyber​​netics，IEEE Transactions on，6，pp 769-772,2010。

[2] ：I. Mani，I. Zhang。“kNN不平衡数据分布的方法：涉及信息提取的案例研究”，2003年在不平衡数据集学习研讨会论文集。

[3] ：体育哈特，“凝聚最近邻规则，”在信息理论，IEEE交易，第一卷。14（3），第515-516页，1968。

[4] ：M. Kubat，S. Matwin，“解决不平衡训练集的诅咒：单方面的选择”，ICML，vol。97，pp.179-186,1997。

[5] ：J. Laurikkala，“通过平衡阶级分配来改善困难小班的识别”，Springer Berlin Heidelberg，2001年。

[6] ：D. Wilson，“使用编辑数据的最近邻规则的渐近性质”，在IEEE Transactions on Systems，Man，and Cyber​​netrics，vol。2（3），408-421页，1972。

[7] ：D. Smith，Michael R.，Tony Martinez和Christophe Giraud-Carrier。“数据复杂性的实例级分析”机器学习95.2（2014）：225-256。

[8] ：NV Chawla，KW Bowyer，LOHall，WP Kegelmeyer，“SMOTE：synthetic minority over-sampling technique，”Journal of artificial intelligence research，321-357,2002。

[9] ：H. Han，W。Wen-Yuan，M. Bing-Huan，“边界线SMOTE：一种新的过采样方法在不平衡数据集学习”，智能计算进展，878-887,2005。

[10] ：HM Nguyen，EW Cooper，K. Kamei，“不平衡数据分类的边界过采样”，国际知识工程和软数据范例，3（1），pp.4-21,2001。

[11] ：G. Batista，RC Prati，MC Monard。“一个平衡机器学习训练数据的几种方法的行为的研究，”ACM Sigkdd探索通讯6（1），20-29,2004。

[12] ：G.Batista，B.Bazzan，M.Monard，“平衡关键词自动标注的训练数据：一个案例研究”，WOB，10-18,2003。

[13] （1，2）：XY刘，吴J.和ZH周，“探索欠了类不平衡学习，”在系统，人在IEEE Transactions与控制，B部分（控制论），第二卷。39，没有。2，第539-550页，2009年4月。

[14] （1，2）：I.托梅克，“在编辑的最近邻规则的实验，”系统，人，和控制论，第一卷IEEE交易。6（6），第448-452页，1976年6月。

[15] ：他，海博，杨白，爱德华加西亚，李树涛。“ADASYN：自适应合成采样方法的不平衡学习，”在IEEE国际神经网络联合会议（IEEE世界大会的计算智能），页1322-1328,2008年。

References:

[1] : I. Tomek, “Two modifications of CNN,” In Systems, Man, and Cybernetics, IEEE Transactions on, vol. 6, pp 769-772, 2010.

[2] : I. Mani, I. Zhang. “kNN approach to unbalanced data distributions: a case study involving information extraction,” In Proceedings of workshop on learning from imbalanced datasets, 2003.

[3] : P. Hart, “The condensed nearest neighbor rule,” In Information Theory, IEEE Transactions on, vol. 14(3), pp. 515-516, 1968.

[4] : M. Kubat, S. Matwin, “Addressing the curse of imbalanced training sets: one-sided selection,” In ICML, vol. 97, pp. 179-186, 1997.

[5] : J. Laurikkala, “Improving identification of difficult small classes by balancing class distribution,” Springer Berlin Heidelberg, 2001.

[6] : D. Wilson, “Asymptotic Properties of Nearest Neighbor Rules Using Edited Data,” In IEEE Transactions on Systems, Man, and Cybernetrics, vol. 2 (3), pp. 408-421, 1972.

[7] : D. Smith, Michael R., Tony Martinez, and Christophe Giraud-Carrier. “An instance level analysis of data complexity.” Machine learning 95.2 (2014): 225-256.

[8] : N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.

[9] : H. Han, W. Wen-Yuan, M. Bing-Huan, “Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning,” Advances in intelligent computing, 878-887, 2005.

[10] : H. M. Nguyen, E. W. Cooper, K. Kamei, “Borderline over-sampling for imbalanced data classification,” International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), pp.4-21, 2001.

[11] : G. Batista, R. C. Prati, M. C. Monard. “A study of the behavior of several methods for balancing machine learning training data,” ACM Sigkdd Explorations Newsletter 6 (1), 20-29, 2004.

[12] : G. Batista, B. Bazzan, M. Monard, [“Balancing Training Data for Automated Annotation of Keywords: a Case Study,” In WOB, 10-18, 2003.

[13] (1, 2) : X. Y. Liu, J. Wu and Z. H. Zhou, “Exploratory Undersampling for Class-Imbalance Learning,” in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 2, pp. 539-550, April 2009.

[14] (1, 2) : I. Tomek, “An Experiment with the Edited Nearest-Neighbor Rule,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 6(6), pp. 448-452, June 1976.

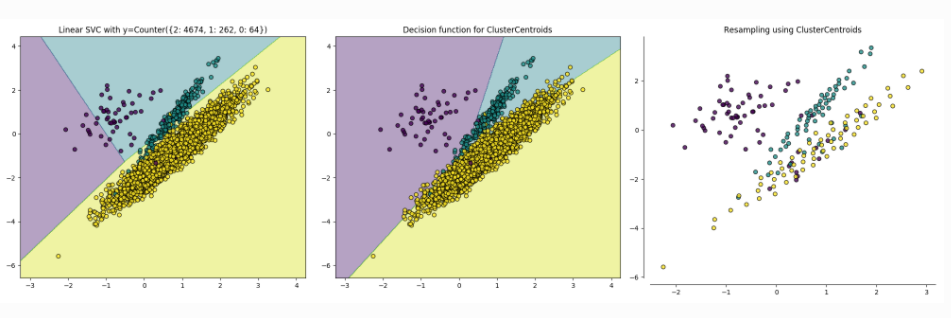
[15] : He, Haibo, Yang Bai, Edwardo A. Garcia, and Shutao Li. “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” In IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322-1328, 2008.

# **不平衡学习imbalanced-learn API**

这是不平衡学习工具箱的完整API文档。

## **[imblearn.under\_sampling](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.under_sampling" \o "imblearn.under_sampling)：欠采样方法**

在**[imblearn.under\_sampling](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.under_sampling" \o "imblearn.under_sampling)**提供的方法下的试样的数据集。



### **原型生成**

该**[imblearn.under\_sampling.prototype\_generation](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.under_sampling.prototype_generation" \o "imblearn.under_sampling.prototype_generation)**子模块包含了为了平衡数据集生成新样本的方法。

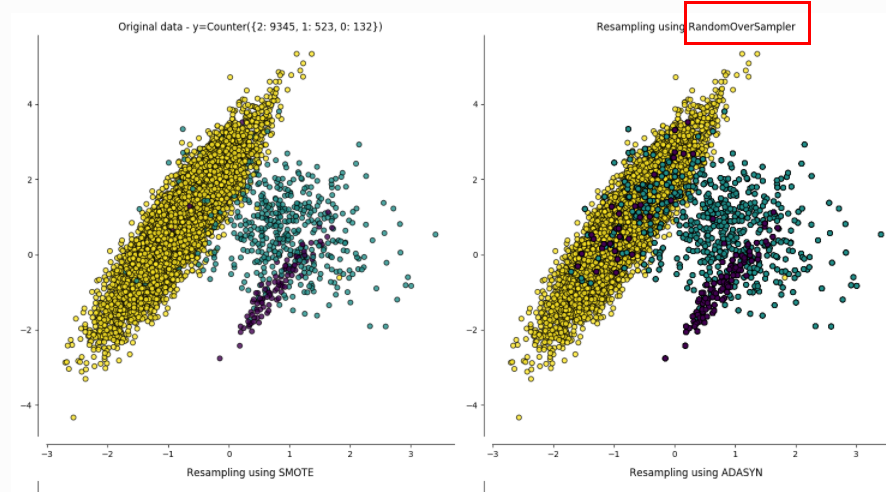
|  |  |
| --- | --- |
| **[under\_sampling.ClusterCentroids](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.ClusterCentroids.html" \l "imblearn.under_sampling.ClusterCentroids" \o "imblearn.under_sampling.ClusterCentroids)**（[比例，...]） | 通过基于聚类方法生成质心来执行欠采样。 |

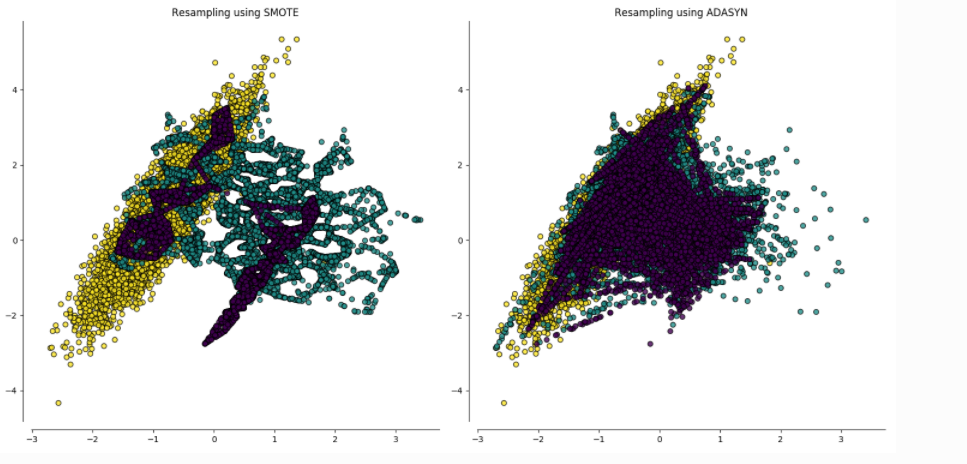
### **原型选择**

该**[imblearn.under\_sampling.prototype\_selection](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.under_sampling.prototype_selection" \o "imblearn.under_sampling.prototype_selection)**子模块包含了为了平衡数据集选择样本的方法。

|  |  |
| --- | --- |
| **[under\_sampling.CondensedNearestNeighbour](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.CondensedNearestNeighbour.html" \l "imblearn.under_sampling.CondensedNearestNeighbour" \o "imblearn.under_sampling.CondensedNearestNeighbour)**（[...]） | 基于压缩最近邻法进行欠采样的类。 |
| **[under\_sampling.EditedNearestNeighbours](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.EditedNearestNeighbours.html" \l "imblearn.under_sampling.EditedNearestNeighbours" \o "imblearn.under_sampling.EditedNearestNeighbours)**（[...]） | 根据编辑的最近邻方法执行欠采样的类。 |
| **[under\_sampling.RepeatedEditedNearestNeighbours](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.RepeatedEditedNearestNeighbours.html" \l "imblearn.under_sampling.RepeatedEditedNearestNeighbours" \o "imblearn.under_sampling.RepeatedEditedNearestNeighbours)**（[...]） | 根据重复编辑的最近邻居方法执行欠采样的类。 |
| **[under\_sampling.AllKNN](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.AllKNN.html" \l "imblearn.under_sampling.AllKNN" \o "imblearn.under_sampling.AllKNN)**（[比例，...]） | 根据AllKNN方法执行欠采样的类。 |
| **[under\_sampling.InstanceHardnessThreshold](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.InstanceHardnessThreshold.html" \l "imblearn.under_sampling.InstanceHardnessThreshold" \o "imblearn.under_sampling.InstanceHardnessThreshold)**（[...]） | 根据实例硬度阈值执行欠采样的类。 |
| **[under\_sampling.NearMiss](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.NearMiss.html" \l "imblearn.under_sampling.NearMiss" \o "imblearn.under_sampling.NearMiss)**（[比例，...]） | 根据NearMiss方法执行欠采样的类。 |
| **[under\_sampling.NeighbourhoodCleaningRule](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.NeighbourhoodCleaningRule.html" \l "imblearn.under_sampling.NeighbourhoodCleaningRule" \o "imblearn.under_sampling.NeighbourhoodCleaningRule)**（[...]） | 根据邻居清理规则执行欠采样的类。 |
| **[under\_sampling.OneSidedSelection](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.OneSidedSelection.html" \l "imblearn.under_sampling.OneSidedSelection" \o "imblearn.under_sampling.OneSidedSelection)**（[比例，...]） | 基于片面选择方法执行欠采样的类。 |
| **[under\_sampling.RandomUnderSampler](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.RandomUnderSampler.html" \l "imblearn.under_sampling.RandomUnderSampler" \o "imblearn.under_sampling.RandomUnderSampler)**（[比例，...]） | 类执行随机欠采样。 |
| **[under\_sampling.TomekLinks](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.TomekLinks.html" \l "imblearn.under_sampling.TomekLinks" \o "imblearn.under_sampling.TomekLinks)**（[比例，...]） | 通过删除Tomek的链接来执行欠采样的类。 |

## **[imblearn.over\_sampling](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.over_sampling" \o "imblearn.over_sampling)：过度采样方法**





在**[imblearn.over\_sampling](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.over_sampling" \o "imblearn.over_sampling)**提供了一组方法来执行过采样的。

|  |  |
| --- | --- |
| **[over\_sampling.ADASYN](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.over_sampling.ADASYN.html" \l "imblearn.over_sampling.ADASYN" \o "imblearn.over_sampling.ADASYN)**（[ratio，random\_state，...]） | 使用ADASYN执行过采样。 |
| **[over\_sampling.RandomOverSampler](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.over_sampling.RandomOverSampler.html" \l "imblearn.over_sampling.RandomOverSampler" \o "imblearn.over_sampling.RandomOverSampler)**（[比例，...]） | 类执行随机过采样。 |
| **[over\_sampling.SMOTE](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.over_sampling.SMOTE.html" \l "imblearn.over_sampling.SMOTE" \o "imblearn.over_sampling.SMOTE)**（[ratio，random\_state，...]） | 使用SMOTE执行过采样的类。 |

## **[imblearn.combine](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.combine" \o "imblearn.combine)：结合了过度采样和欠采样方法**

在**[imblearn.combine](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.combine" \o "imblearn.combine)**提供了结合过采样和欠采样方法。

|  |  |
| --- | --- |
| **[combine.SMOTEENN](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.combine.SMOTEENN.html" \l "imblearn.combine.SMOTEENN" \o "imblearn.combine.SMOTEENN)**（[ratio，random\_state，...]） | 使用SMOTE执行过采样并使用ENN进行清洁的类。 |
| **[combine.SMOTETomek](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.combine.SMOTETomek.html" \l "imblearn.combine.SMOTETomek" \o "imblearn.combine.SMOTETomek)**（[ratio，random\_state，...]） | 类使用SMOTE执行过采样并使用Tomek链接进行清理。 |

## **[imblearn.ensemble](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.ensemble" \o "imblearn.ensemble)：集成方法**

该**[imblearn.ensemble](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.ensemble" \o "imblearn.ensemble)**模块包括生成组合在集合内的欠采样子集的方法。

|  |  |
| --- | --- |
| **[ensemble.BalanceCascade](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.ensemble.BalanceCascade.html" \l "imblearn.ensemble.BalanceCascade" \o "imblearn.ensemble.BalanceCascade)**（[比例，...]） | 通过使用估计器迭代地对不平衡数据集进行欠采样来创建平衡集的集合。 |
| **[ensemble.BalancedBaggingClassifier](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.ensemble.BalancedBaggingClassifier.html" \l "imblearn.ensemble.BalancedBaggingClassifier" \o "imblearn.ensemble.BalancedBaggingClassifier)**（[...]） | 带有额外平衡的装袋分类器。 |
| **[ensemble.EasyEnsemble](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.ensemble.EasyEnsemble.html" \l "imblearn.ensemble.EasyEnsemble" \o "imblearn.ensemble.EasyEnsemble)**（[比例，...]） | 通过迭代应用随机欠采样来创建集合集。 |

## **[imblearn.pipeline](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.pipeline" \o "imblearn.pipeline)：管道**

该**[imblearn.pipeline](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.pipeline" \o "imblearn.pipeline)**模块实现实用程序来构建一个复合估计器，作为变换链，样本和估计器。

|  |  |
| --- | --- |
| **[pipeline.Pipeline](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.pipeline.Pipeline.html" \l "imblearn.pipeline.Pipeline" \o "imblearn.pipeline.Pipeline)**（步骤[，内存]） | 使用最终估计器进行变换和重新采样的流水线。 |

|  |  |
| --- | --- |
| **[pipeline.make\_pipeline](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.pipeline.make_pipeline.html" \l "imblearn.pipeline.make_pipeline" \o "imblearn.pipeline.make_pipeline)**（\*脚步） | 从给定的估计者构建管道。 |

## **[imblearn.metrics](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.metrics" \o "imblearn.metrics)：指标**

该**[imblearn.metrics](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.metrics" \o "imblearn.metrics)**模块包括分数函数，性能指标和成对指标和距离计算。

|  |  |
| --- | --- |
| **[metrics.classification\_report\_imbalanced](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.metrics.classification_report_imbalanced.html" \l "imblearn.metrics.classification_report_imbalanced" \o "imblearn.metrics.classification_report_imbalanced)**（......） | 根据与不平衡相关的指标构建分类报告 |
| **[metrics.sensitivity\_specificity\_support](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.metrics.sensitivity_specificity_support.html" \l "imblearn.metrics.sensitivity_specificity_support" \o "imblearn.metrics.sensitivity_specificity_support)**（......） | 计算每个班级的敏感度，特异度和支持度 |
| **[metrics.sensitivity\_score](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.metrics.sensitivity_score.html" \l "imblearn.metrics.sensitivity_score" \o "imblearn.metrics.sensitivity_score)**（y\_true，y\_pred [，...]） | 计算灵敏度 |
| **[metrics.specificity\_score](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.metrics.specificity_score.html" \l "imblearn.metrics.specificity_score" \o "imblearn.metrics.specificity_score)**（y\_true，y\_pred [，...]） | 计算特异性 |
| **[metrics.geometric\_mean\_score](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.metrics.geometric_mean_score.html" \l "imblearn.metrics.geometric_mean_score" \o "imblearn.metrics.geometric_mean_score)**（y\_true，y\_pred） | 计算几何平均值 |
| **[metrics.make\_index\_balanced\_accuracy](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.metrics.make_index_balanced_accuracy.html" \l "imblearn.metrics.make_index_balanced_accuracy" \o "imblearn.metrics.make_index_balanced_accuracy)**（[...]） | 使用索引平衡准确性平衡任何得分功能 |

## **[imblearn.datasets](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.datasets" \o "imblearn.datasets)：数据集**

在**[imblearn.datasets](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.datasets" \o "imblearn.datasets)**提供的方法来产生不平衡数据。

|  |  |
| --- | --- |
| **[datasets.make\_imbalance](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.datasets.make_imbalance.html" \l "imblearn.datasets.make_imbalance" \o "imblearn.datasets.make_imbalance)**（X，y，比率[，...]） | 将数据集转换为特定比例的不平衡数据集。 |
| **[datasets.fetch\_datasets](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.datasets.fetch_datasets.html" \l "imblearn.datasets.fetch_datasets" \o "imblearn.datasets.fetch_datasets)**（[data\_home，...]） | 从Zenodo加载基准数据集，如果需要的话下载。 |

## **[imblearn.utils](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.utils" \o "imblearn.utils)：公用事业**

该**[imblearn.utils](http://contrib.scikit-learn.org/imbalanced-learn/stable/api.html" \l "module-imblearn.utils" \o "imblearn.utils)**模块包含各种实用程序。

|  |  |
| --- | --- |
| **utils.estimator\_checks.check\_estimator** |  |
| **[utils.check\_neighbors\_object](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.utils.check_neighbors_object.html" \l "imblearn.utils.check_neighbors_object" \o "imblearn.utils.check_neighbors_object)**（nn\_name，nn\_object） | 检查对象是一致的是一个NN。 |
| **[utils.check\_ratio](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.utils.check_ratio.html" \l "imblearn.utils.check_ratio" \o "imblearn.utils.check_ratio)**（ratio，y，sampling\_type，...） | 采样器的比率验证。 |
| **[utils.hash\_X\_y](http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.utils.hash_X_y.html" \l "imblearn.utils.hash_X_y" \o "imblearn.utils.hash_X_y)**（X，y [，n\_samples]） | 计算输入数组的哈希值。 |