

# Package ‘scutr’

October 14, 2022

**Title** Balancing Multiclass Datasets for Classification Tasks

**Version** 0.1.2

**Maintainer** Keenan Ganz <ganzkeenan1@gmail.com>

## Description

Imbalanced training datasets impede many popular classifiers. To balance training data, a combination of oversampling minority classes and undersampling majority classes is useful. This package implements the SCUT (SMOTE and Cluster-based Undersampling Technique) algorithm as described in Agrawal et. al. (2015) <doi:10.5220/0005595502260234>. Their paper uses model-based clustering and synthetic oversampling to balance multiclass training datasets, although other resampling methods are provided in this package.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** smotefamily, parallel, mclust

**Depends** R (>= 2.10)

**URL** <https://github.com/s-kganz/scutr>

**BugReports** <https://github.com/s-kganz/scutr/issues>

**Suggests** testthat (>= 2.0.0)

**Config/testthat/edition** 2

**NeedsCompilation** no

**Author** Keenan Ganz [aut, cre]

**Repository** CRAN

**Date/Publication** 2021-06-24 11:40:02 UTC

## R topics documented:

bullseye . . . . .	2
imbalance . . . . .	2

oversample_smote . . . . .	3
resample_random . . . . .	4
sample_classes . . . . .	4
SCUT . . . . .	5
undersample_hclust . . . . .	7
undersample_kmeans . . . . .	8
undersample_mclust . . . . .	8
undersample_mindist . . . . .	9
undersample_tomek . . . . .	10
validate_dataset . . . . .	11
wine . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

bullseye	<i>An imbalanced dataset with a minor class centered around the origin with a majority class surrounding the center.</i>
----------	--

---

### Description

An imbalanced dataset with a minor class centered around the origin with a majority class surrounding the center.

### Usage

```
bullseye
```

### Format

a data.frame with 1000 rows and 3 columns.

### Source

<https://gist.github.com/s-k ganz/c2534666e369f8e19491bb29d53c619d>

---

imbalance	<i>An imbalanced dataset with randomly placed normal distributions around the origin. The nth class has n * 10 observations.</i>
-----------	--

---

### Description

An imbalanced dataset with randomly placed normal distributions around the origin. The nth class has n \* 10 observations.

### Usage

```
imbalance
```

**Format**

a data.frame with 2100 rows and 11 columns

**Source**

<https://gist.github.com/s-kganzt/d08473f9492d48ea0e56c3c8a3fe1a74>

---

oversample_smote	<i>Oversample a dataset by SMOTE.</i>
------------------	---------------------------------------

---

**Description**

Oversample a dataset by SMOTE.

**Usage**

```
oversample_smote(data, cls, cls_col, m)
```

**Arguments**

data	Dataset to be oversampled.
cls	Class to be oversampled.
cls_col	Column containing class information.
m	Desired number of samples in the oversampled data.

**Value**

The oversampled dataset.

**Examples**

```
table(iris$Species)
smoted <- oversample_smote(iris, "setosa", "Species", 100)
nrow(smoted)
```

---

resample_random	<i>Randomly resample a dataset.</i>
-----------------	-------------------------------------

---

### Description

This function is used to resample a dataset by randomly removing or duplicating rows. It is usable for both oversampling and undersampling.

### Usage

```
resample_random(data, cls, cls_col, m)
```

### Arguments

data	Dataframe to be resampled.
cls	Class that should be randomly resampled.
cls_col	Column containing class information.
m	Desired number of samples.

### Value

Resampled dataframe containing only cls.

### Examples

```
set.seed(1234)
only2 <- resample_random(wine, 2, "type", 15)
```

---

sample_classes	<i>Stratified index sample of different values in a vector.</i>
----------------	---

---

### Description

Stratified index sample of different values in a vector.

### Usage

```
sample_classes(vec, tot_sample)
```

### Arguments

vec	Vector of values to sample from.
tot_sample	Total number of samples.

**Value**

A vector of indices that can be used to select a balanced population of values from `vec`.

**Examples**

```
vec <- sample(1:5, 30, replace = TRUE)
table(vec)
sample_ind <- sample_classes(vec, 15)
table(vec[sample_ind])
```

---

SCUT

*SMOTE and cluster-based undersampling technique.*

---

**Description**

This function balances multiclass training datasets. In a dataframe with  $n$  classes and  $m$  rows, the resulting dataframe will have  $m / n$  rows per class. `SCUT_parallel()` distributes each over/undersampling task across multiple cores. Speedup usually occurs only if there are many classes using one of the slower resampling techniques (e.g. `undersample_mclust()`). Note that `SCUT_parallel()` will always run on one core on Windows.

**Usage**

```
SCUT(
  data,
  cls_col,
  oversample = oversample_smote,
  undersample = undersample_mclust,
  osamp_opts = list(),
  usamp_opts = list()
)
```

```
SCUT_parallel(
  data,
  cls_col,
  ncores = detectCores()%%2,
  oversample = oversample_smote,
  undersample = undersample_mclust,
  osamp_opts = list(),
  usamp_opts = list()
)
```

**Arguments**

<code>data</code>	Numeric data frame.
<code>cls_col</code>	The column in <code>data</code> with class membership.

oversample	Oversampling method. Must be a function with the signature <code>foo(data, cls, cls_col, m, ...)</code> that returns a data frame, one of the <code>oversample_*</code> functions, or <code>resample_random()</code> .
undersample	Undersampling method. Must be a function with the signature <code>foo(data, cls, cls_col, m, ...)</code> that returns a data frame, one of the <code>undersample_*</code> functions, or <code>resample_random()</code> .
osamp_opts	List of options passed to the oversampling function.
usamp_opts	List of options passed to the undersampling function.
ncores	Number of cores to use with <code>SCUT_parallel()</code> .

### Details

Custom functions can be used to perform under/oversampling (see the required signature below). Parameters represented by `...` should be passed via `osamp_opts` or `usamp_opts` as a list.

### Value

A dataframe with equal class distribution.

### References

Agrawal A, Viktor HL, Paquet E (2015). 'SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling.' In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, volume 01, 226-234.

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002). 'SMOTE: Synthetic Minority Over-sampling Technique.' *Journal of Artificial Intelligence Research*, 16, 321-357. ISSN 1076-9757, doi: [10.1613/jair.953](https://doi.org/10.1613/jair.953), <https://www.jair.org/index.php/jair/article/view/10302>.

### Examples

```
ret <- SCUT(iris, "Species", undersample = undersample_hclust,
           usamp_opts = list(dist_calc="manhattan"))
ret2 <- SCUT(chickwts, "feed", undersample = undersample_kmeans)
table(ret$Species)
table(ret2$feed)
# SCUT_parallel fires a warning if ncores > 1 on Windows and will run on
# one core only.
ret <- SCUT_parallel(wine, "type", ncores = 1, undersample = undersample_kmeans)
table(ret$type)
```

---

undersample\_hclust      *Undersample a dataset by hierarchical clustering.*

---

## Description

Undersample a dataset by hierarchical clustering.

## Usage

```
undersample_hclust(  
  data,  
  cls,  
  cls_col,  
  m,  
  k = 5,  
  h = NA,  
  dist_calc = "euclidean"  
)
```

## Arguments

data	Dataset to be undersampled.
cls	Majority class that will be undersampled.
cls_col	Column in data containing class memberships.
m	Number of samples in undersampled dataset.
k	Number of clusters to derive from clustering.
h	Height at which to cut the clustering tree. k must be NA for this to be used.
dist_calc	Distance calculation method. See <a href="#">dist()</a> .

## Value

Undersampled dataframe containing only cls.

## Examples

```
table(iris$Species)  
undersamp <- undersample_hclust(iris, "setosa", "Species", 15)  
nrow(undersamp)
```

---

undersample\_kmeans      *Undersample a dataset by kmeans clustering.*

---

**Description**

Undersample a dataset by kmeans clustering.

**Usage**

```
undersample_kmeans(data, cls, cls_col, m, k = 5)
```

**Arguments**

data	Dataset to be undersampled.
cls	Class to be undersampled.
cls_col	Column containing class information.
m	Number of samples in undersampled dataset.
k	Number of centers in clustering.

**Value**

The undersampled dataframe containing only instances of cls.

**Examples**

```
table(iris$Species)
undersamp <- undersample_kmeans(iris, "setosa", "Species", 15)
nrow(undersamp)
```

---

undersample\_mclust      *Undersample a dataset by expectation-maximization clustering*

---

**Description**

Undersample a dataset by expectation-maximization clustering

**Usage**

```
undersample_mclust(data, cls, cls_col, m)
```

**Arguments**

data	Data to be undersampled.
cls	Class to be undersampled.
cls_col	Class column.
m	Number of samples in undersampled dataset.



**Value**

The undersampled dataframe containing only instance of `cls`.

**Examples**

```
setosa <- iris[iris$Species == "setosa", ]
nrow(setosa)
undersamp <- undersample_mclust(setosa, "setosa", "Species", 15)
nrow(undersamp)
```

---

`undersample_mindist`     *Undersample a dataset by iteratively removing the observation with the lowest total distance to its neighbors of the same class.*

---

**Description**

Undersample a dataset by iteratively removing the observation with the lowest total distance to its neighbors of the same class.

**Usage**

```
undersample_mindist(data, cls, cls_col, m, dist_calc = "euclidean")
```

**Arguments**

<code>data</code>	Dataset to undersample. Aside from <code>cls_col</code> , must be numeric.
<code>cls</code>	Class to be undersampled.
<code>cls_col</code>	Column containing class information.
<code>m</code>	Desired number of observations after undersampling.
<code>dist_calc</code>	Method for distance calculation. See <code>dist()</code> .

**Value**

An undersampled dataframe.

**Examples**

```
setosa <- iris[iris$Species == "setosa", ]
nrow(setosa)
undersamp <- undersample_mindist(setosa, "setosa", "Species", 50)
nrow(undersamp)
```

---

undersample\_tomek      *Undersample a dataset by removing Tomek links.*

---

### Description

A Tomek link is a minority instance and majority instance that are each other's nearest neighbor. This function removes sufficient Tomek links that are an instance of `cls` to yield `m` instances of `cls`. If desired, samples are randomly discarded to yield `m` rows if insufficient Tomek links are in the data.

### Usage

```
undersample_tomek(
  data,
  cls,
  cls_col,
  m,
  tomek = "minor",
  force_m = TRUE,
  dist_calc = "euclidean"
)
```

### Arguments

<code>data</code>	Dataset to be undersampled.
<code>cls</code>	Majority class to be undersampled.
<code>cls_col</code>	Column in data containing class memberships.
<code>m</code>	Desired number of samples in undersampled dataset.
<code>tomek</code>	Definition used to determine if a point is considered a minority in the Tomek link definition. <ul style="list-style-type: none"> <li>• <code>minor</code>: Minor classes are all those with fewer than <code>m</code> instances.</li> <li>• <code>diff</code>: Minor classes are all those that aren't <code>cls</code>.</li> </ul>
<code>force_m</code>	If TRUE, uses random undersampling to discard samples if insufficient Tomek links are present to yield <code>m</code> rows of data.
<code>dist_calc</code>	Distance calculation method. See <a href="#">dist()</a> .

### Value

Undersampled dataframe containing only `cls`.

### Examples

```
table(iris$Species)
undersamp <- undersample_tomek(iris, "setosa", "Species", 15, tomek = "diff", force_m = TRUE)
nrow(undersamp)
undersamp2 <- undersample_tomek(iris, "setosa", "Species", 15, tomek = "diff", force_m = FALSE)
nrow(undersamp2)
```

---

validate_dataset	<i>Validate a dataset for resampling.</i>
------------------	---

---

**Description**

This functions checks that the given column is present in the data and that all columns besides the class column are numeric.

**Usage**

```
validate_dataset(data, cls_col)
```

**Arguments**

data	Dataframe to validate.
cls_col	Column with class information.

**Value**

NA

---

wine	<i>Type and chemical analysis of three different kinds of wine.</i>
------	---

---

**Description**

Type and chemical analysis of three different kinds of wine.

**Usage**

```
wine
```

**Format**

a data.frame with 178 rows and 14 columns

**Source**

<https://archive.ics.uci.edu/ml/datasets/Wine>

# Index

## \* datasets

- bullseye, [2](#)
- imbalance, [2](#)
- wine, [11](#)

bullseye, [2](#)

dist, [7](#), [9](#), [10](#)

imbalance, [2](#)

oversample\_smote, [3](#)

resample\_random, [4](#), [6](#)

sample\_classes, [4](#)

SCUT, [5](#)

SCUT\_parallel, [5](#), [6](#)

SCUT\_parallel (SCUT), [5](#)

undersample\_hclust, [7](#)

undersample\_kmeans, [8](#)

undersample\_mclust, [5](#), [8](#)

undersample\_mindist, [9](#)

undersample\_tomek, [10](#)

validate\_dataset, [11](#)

wine, [11](#)