Package ‘cattonum’

June 15, 2020

Type Package
Title Encode Categorical Features
Version 0.0.5
Maintainer Bernie Gray <bfgray3@gmail.com>
Description Functions and S3 classes for the following methods of encoding categorical features as numerics: aggregate, dummy, frequency, label, leave-one-out, mean, median, and one-hot.
License MIT + file LICENSE
URL https://github.com/bfgray3/cattonum
BugReports https://github.com/bfgray3/cattonum/issues
Depends R (>= 3.6.0)
Imports dplyr (>= 1.0.0), purrr, rlang (>= 0.4.6), stats, tibble (>= 2.1.3), tidyselect (>= 1.1.0), Rcpp
Suggests covr, knitr, lintr, nycflights13, ranger, rmarkdown, roxygen2, testthat
LinkingTo Rcpp
VignetteBuilder knitr
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation yes
Author Bernie Gray [aut, cre] (<https://orcid.org/0000-0001-9190-6032>),
Mark Roepke [ctb]
Repository CRAN
Date/Publication 2020-06-15 04:50:06 UTC
cattonum: Encode Categorical Features

Functions for dummy encoding, frequency encoding, label encoding, leave-one-out encoding, mean encoding, median encoding, and one-hot encoding.

Constructor for class cattonum_df

Usage

```r
cattonum_df(x = NULL)
```

Arguments

- `x` NULL (the default), or a tibble or data.frame.

Value

Either an empty data frame (if `x` is NULL), or `x`. In both cases, the class is `c("cattonum_df","data.frame")`.

Examples

```
cattonum_df(iris)
cattonum_df()
```
cattonum_df2  Constructor for class cattonum_df2

Description

Constructor for class cattonum_df2

Usage

cattonum_df2(train = NULL, test = NULL)

Arguments

- **train**: NULL (the default), or a tibble or data.frame.
- **test**: NULL (the default), or a tibble or data.frame with the same names as **train**.

Value

A list of class cattonum_df2 with names "train" and "test".

Examples

```r
cattonum_df2()
```

catto_aggregate  Aggregate function encoding

Description

Aggregate function encoding

Usage

```r
catto_aggregate(
  train,
  ...,
  aggregate_fun,
  response = NULL,
  test = NULL,
  verbose = TRUE
)
```
Arguments

- **train**: The training data, in a data.frame or tibble.
- **...**: The columns to be encoded. If none are specified, then all character and factor columns are encoded.
- **aggregate_fun**: The aggregate function to be applied to the response variable for the rows belonging to the relevant level of the categorical predictor. Takes a vector and returns a length one vector.
- **response**: The response variable used to calculate aggregate summaries.
- **test**: The test data, in a data.frame or tibble.
- **verbose**: Should informative messages be printed? Defaults to TRUE.

Value

The encoded dataset in a *cattonum_df* if no test dataset was provided, and the encoded datasets in a *cattonum_df2* otherwise.

Examples

```r
catto_aggregate(iris, aggregate_fun = max, response = Sepal.Length)
```

catto_dummy(iris)

Description

Dummy encoding

Usage

```r
catto_dummy(train, ..., test, verbose = TRUE)
```

Arguments

- **train**: The training data, in a data.frame or tibble.
- **...**: The columns to be encoded. If none are specified, then all character and factor columns are encoded.
- **test**: The test data, in a data.frame or tibble.
- **verbose**: Should informative messages be printed? Defaults to TRUE (not yet used).

Value

The encoded dataset in a *cattonum_df* if no test dataset was provided, and the encoded datasets in a *cattonum_df2* otherwise.

Examples

```r
catto_dummy(iris)
```
**catto_freq**

*Frequency encoding*

**Description**

Frequency encoding

**Usage**

```r
catto_freq(train, ..., test, verbose = TRUE)
```

**Arguments**

- `train`: The training data, in a `data.frame` or `tibble`.
- `...`: The columns to be encoded. If none are specified, then all character and factor columns are encoded.
- `test`: The test data, in a `data.frame` or `tibble`.
- `verbose`: Should informative messages be printed? Defaults to `TRUE` (not yet used).

**Value**

The encoded dataset in a `cattonum_df` if no test dataset was provided, and the encoded datasets in a `cattonum_df2` otherwise.

**Examples**

```r
catto_freq(iris)
```

---

**catto_label**

*Label encoding*

**Description**

Label encoding

**Usage**

```r
catto_label(train, ..., test, ordering = "increasing", verbose = TRUE)
```

---
Arguments

- **train**: The training data, in a `data.frame` or `tibble`.
- **...**: The columns to be encoded. If none are specified, then all character and factor columns are encoded.
- **test**: The test data, in a `data.frame` or `tibble`.
- **ordering**: How should labels be assigned to levels? There are three different ways to pass this argument. First, a length one character vector with value "increasing", "decreasing", "observed", or "random" will apply that ordering to each column being encoded. Second, a character vector of length greater than one may be passed, specifying one of the above four options for each column being encoded. Finally, a list may be passed specifying a user-defined ordering for each column being encoded.
- **verbose**: Should informative messages be printed? Defaults to `TRUE` (not yet used).

Value

The encoded dataset in a `cattonum_df` if no test dataset was provided, and the encoded datasets in a `cattonum_df2` otherwise.

Examples

```r
catto_label(iris)

y <- 2^0:5
x1 <- c("a", "b", NA, "b", "a", "a")
x2 <- c("c", "c", "c", "d", "d", "c")
df_fact <- data.frame(y, x1, x2)

catto_label(df_fact, 
  ordering = list(c("b", "a"), c("c", "d"))
)
catto_label(df_fact, ordering = c("increasing", "decreasing"))
```

---

**catto_loo**  
*Leave-one-out encoding*

Description

Leave-one-out encoding

Usage

```r
catto_loo(train, ..., response = NULL, test = NULL, verbose = TRUE)
```
**catto_mean**

**Arguments**

- **train**: The training data, in a data.frame or tibble.
- **...**: The columns to be encoded. If none are specified, then all character and factor columns are encoded.
- **response**: The response variable used to calculate means.
- **test**: The test data, in a data.frame or tibble.
- **verbose**: Should informative messages be printed? Defaults to TRUE.

**Value**

The encoded dataset in a cattonum_df if no test dataset was provided, and the encoded datasets in a cattonum_df2 otherwise.

**Examples**

```
catto_loo(iris, response = Sepal.Length)
```
### cattot_median

**Median encoding**

**Description**

Median encoding

**Usage**

```r
cattot_median(train, ..., response = NULL, test = NULL, verbose = TRUE)
```

**Arguments**

- `train`: The training data, in a `data.frame` or `tibble`.
- `...`: The columns to be encoded. If none are specified, then all character and factor columns are encoded.
- `response`: The response variable used to calculate medians.
- `test`: The test data, in a `data.frame` or `tibble`.
- `verbose`: Should informative messages be printed? Defaults to `TRUE`.

**Value**

The encoded dataset in a `cattonum_df` if no test dataset was provided, and the encoded datasets in a `cattonum_df2` otherwise.

**Examples**

```r
cattot_median(iris, response = Sepal.Length)
```

### cattot_onehot

**One-hot encoding**

**Description**

One-hot encoding

**Usage**

```r
cattot_onehot(train, ..., test, verbose = TRUE)
```
Arguments

train
  The training data, in a data.frame or tibble.
...
  The columns to be encoded. If none are specified, then all character and factor
columns are encoded.
test
  The test data, in a data.frame or tibble.
verbose
  Should informative messages be printed? Defaults to TRUE (not yet used).

Value

The encoded dataset in a cattonum_df if no test dataset was provided, and the encoded datasets in
a cattonum_df2 otherwise.

Examples

catto_onehot(iris)
Index

catto_aggregate, 3
catto_dummy, 4
catto_freq, 5
catto_label, 5
catto_loo, 6
catto_mean, 7
catto_median, 8
catto_onehot, 8
cattonum, 2
cattonum_df, 2
cattonum_df2, 3