Package ‘autogam’

October 2, 2024

Title Automate the Creation of Generalized Additive Models (GAMs)

Version 0.0.1

Language en-US

Description This wrapper package for 'mgcv' makes it easier to create high-performing Generalized Additive Models (GAMs). With its central function autogam(), by entering just a dataset and the name of the outcome column as inputs, 'AutoGAM' tries to automate the procedure of configuring a highly accurate GAM which performs at reasonably high speed, even for large datasets.

License MIT + file LICENSE

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, mgcv, purrr, stats, stringr

URL https://github.com/tripartio/autogam, https://tripartio.github.io/autogam/

BugReports https://github.com/tripartio/autogam/issues

NeedsCompilation no

Author Chitu Okoli [aut, cre] (<https://orcid.org/0000-0001-5574-7572>)

Maintainer Chitu Okoli <Chitu.Okoli@skema.edu>

Repository CRAN

Date/Publication 2024-10-02 13:50:02 UTC

Contents

autogam *Automate the creation of a Generalized Additive Model (GAM)*

Description

`autogam()` is a wrapper for 'mgcv::gam()' that makes it easier to create high-performing Generalized Additive Models (GAMs). By entering just a dataset and the name of the outcome column as inputs, `autogam()` tries to automate the procedure of configuring a highly accurate GAM which performs at reasonably high speed, even for large datasets.

Usage

```
autogam(data, y_col, ...)
```

Arguments

`data`	dataframe. All the variables in `data` will be used to predict `y_col`. To exclude any variables, assign as `data` only the subset of variables desired.
`y_col`	character(1). Name of the y outcome variable.
`...`	Arguments passed on to `mgcv::gam()`.

Value

Returns an `mgcv::gam` object, the result of predicting `y_col` from all other variables in `data`.

Examples

```
autogam(mtcars, 'mpg')
```

smooth_formula_string *Create a character string for a mgcv::gam formula*

Description

Create a character string that wraps appropriate variables in a dataframe with `s()` smooth functions. Based on the datatype of each variable, it determines whether it is a numeric variable to be smoothed:

- Non-numeric: no smoothing.
- Numeric: determine knots based on the number of unique values for that variable:
  - `<= 4`: no smoothing
  - `5 to 19` (inclusive): smooth function with knots equal to the floored half of the number of unique values. E.g., 6 unique values receive 3 knots, 7 will receive 3 knots, and 8 will receive 4 knots.
  - `>= 20`: smooth function with no specified number of knots, allowing the `gam()` function to detect the appropriate number.

Usage

```
smooth_formula_string(data, y_col, smooth_fun = "s", expand_parametric = TRUE)
```

Arguments

`data`	dataframe. All the variables in `data` except `y_col` will be listed in the resulting formula string. To exclude any variables, assign as `data` only the subset of variables desired.
`y_col`	character(1). Name of the y outcome variable.
`smooth_fun`	character(1). Function to use for smooth wraps; default is 's' for the `s()` function.
`expand_parametric`	logical(1). If `TRUE` (default), explicitly list each non-smooth (parametric) term. If `FALSE`, use `.` to lump together all non-smooth terms.

Value

Returns a single character string that represents a formula with `y_col` on the left and all other variables in `data` on the right, each formatted with an appropriate `s()` function when applicable.

Examples

```
smooth_formula_string(mtcars, 'mpg')
```

Index