Package ‘tidypredict’

October 14, 2022

Title Run Predictions Inside the Database

Version 0.4.9

Description It parses a fitted 'R' model object, and returns a formula in 'Tidy Eval' code that calculates the predictions. It works with several databases back-ends because it leverages 'dplyr' and 'dbplyr' for the final 'SQL' translation of the algorithm. It currently supports lm(), glm(), randomForest(), ranger(), earth(), xgb.Booster.complete(), cubist(), and ctree() models.

License MIT + file LICENSE


BugReports https://github.com/tidymodels/tidypredict/issues

Depends R (>= 3.1)

Imports dplyr (>= 0.7), generics, knitr, purrr, rlang, stringr, tibble, tidyr

Suggests covr, Cubist, DBI, dbplyr, earth (>= 5.1.2), methods, mlbench, modeldata, nycflights13, parsnip, partykit, randomForest, ranger, rmarkdown, RSQLite, testthat (>= 3.0.0), xgboost, yaml

VignetteBuilder knitr

Config/Needs/website tidyverse/tidytemplate

Encoding UTF-8

RoxygenNote 7.2.0.9000

Config/testthat/edition 3

NeedsCompilation no

Author Max Kuhn [aut, cre], Edgar Ruiz [aut]

Maintainer Max Kuhn <max@rstudio.com>

Repository CRAN

Date/Publication 2022-05-25 19:20:02 UTC
### R topics documented:

- acceptable_formula
- as_parsed_model
- parse_model
- tidy.pm_regression
- tidypredict_fit
- tidypredict_interval
- tidypredict_test
- tidypredict_to_column

## acceptable_formula

Checks that the formula can be parsed

**Description**

Uses an S3 method to check that a given formula can be parsed based on its class. It currently scans for contrasts that are not supported and in-line functions. (e.g: `lm(wt ~ as.factor(am)))`. Since this function is meant for function interaction, as opposed to human interaction, a successful check is silent.

**Usage**

```r
acceptable_formula(model)
```

**Arguments**

- `model` An R model object

**Examples**

```r
model <- lm(mpg ~ wt, mtcars)
acceptable_formula(model)
```

## as_parsed_model

Prepares parsed model object

**Description**

Prepares parsed model object

**Usage**

```r
as_parsed_model(x)
```
**parse_model**

*Arguments*

- **x**
  A parsed model object

*Description*

It parses a fitted R model’s structure and extracts the components needed to create a dplyr formula for prediction. The function also creates a data frame using a specific format so that other functions in the future can also pass parsed tables to a given formula creating function.

*Usage*

```
parse_model(model)
```

*Arguments*

- **model**
  An R model object.

*Examples*

```
library(dplyr)
df <- mutate(mtcars, cyl = paste0("cyl", cyl))
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = df)
parse_model(model)
```

**tidy.pm_regression**

*Tidy the parsed model results*

*Description*

Tidy the parsed model results

*Usage*

```
## S3 method for class 'pm_regression'
tidy(x, ...)
```

*Arguments*

- **x**
  A parsed_model object
- **...**
  Reserved for future use
tidypredict_fit

Returns a Tidy Eval formula to calculate fitted values.

Description

It parses a model or uses an already parsed model to return a Tidy Eval formula that can then be used inside a dplyr command.

Usage

`tidypredict_fit(model)`

Arguments

- `model`  
  An R model or a list with a parsed model.

Examples

```r
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = mtcars)
tidypredict_fit(model)
```

---

tidypredict_interval

Returns a Tidy Eval formula to calculate prediction interval.

Description

It parses a model or uses an already parsed model to return a Tidy Eval formula that can then be used inside a dplyr command.

Usage

`tidypredict_interval(model, interval = 0.95)`

Arguments

- `model`  
  An R model or a list with a parsed model
- `interval`  
  The prediction interval, defaults to 0.95

Details

The result still has to be added to and subtracted from the fit to obtain the upper and lower bound respectively.
Examples

```r
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = mtcars)
tidypredict_interval(model)
```

**Description**

Compares the results of `predict()` and `tidypredict_to_column()` functions.

**Usage**

```r
tidypredict_test(
  model,
  df = model$model,
  threshold = 1e-12,
  include_intervals = FALSE,
  max_rows = NULL,
  xg_df = NULL
)
```

**Arguments**

- **model**: An R model or a list with a parsed model. It currently supports `lm()`, `glm()` and `randomForest()` models.
- **df**: A data frame that contains all of the needed fields to run the prediction. It defaults to the "model" data frame object inside the model object.
- **threshold**: The number that a given result difference, between `predict()` and `tidypredict_to_column()` should not exceed. For continuous predictions, the default value is `0.000000000001` (`1e-12`), and for categorical predictions, the default value is `0`.
- **include_intervals**: Switch to indicate if the prediction intervals should be included in the test. It defaults to `FALSE`.
- **max_rows**: The number of rows in the object passed in the df argument. Highly recommended for large data sets.
- **xg_df**: A xgb.DMatrix object, required only for XGBoost models. It defaults to `NULL` recommended for large data sets.

**Examples**

```r
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = mtcars)
tidypredict_test(model)
```
tidypredict_to_column  

 Adds the prediction columns to a piped command set.

Description

Adds a new column with the results from tidypredict_fit() to a piped command set. If add_interval is set to TRUE, it will add two additional columns- one for the lower and another for the upper prediction interval bounds.

Usage

```
tidypredict_to_column(
  df,
  model,
  add_interval = FALSE,
  interval = 0.95,
  vars = c("fit", "upper", "lower")
)
```

Arguments

- **df**: A data.frame or tibble
- **model**: An R model or a parsed model inside a data frame
- **add_interval**: Switch that indicates if the prediction interval columns should be added. Defaults to FALSE
- **interval**: The prediction interval, defaults to 0.95. Ignored if add_interval is set to FALSE
- **vars**: The name of the variables that this function will produce. Defaults to "fit", "upper", and "lower".
Index

acceptable_formula, 2
as_parsed_model, 2

parse_model, 3
	tidy.pm_regress, 3
tidypredict_fit, 4
tidypredict_interval, 4
tidypredict_test, 5
tidypredict_to_column, 6