Package ‘tidyAML’

March 12, 2024

Title  Automatic Machine Learning with 'tidymodels'

Version  0.0.5

Description  The goal of this package will be to provide a simple interface for automatic machine learning that fits the 'tidymodels' framework. The intention is to work for regression and classification problems with a simple verb framework.

License  MIT + file LICENSE

Encoding  UTF-8

URL  https://www.spsanderson.com/tidyAML/,
      https://github.com/spsanderson/tidyAML

BugReports  https://github.com/spsanderson/tidyAML/issues

Depends  parsnip, R (>= 4.1.0)

Suggests  knitr, markdown, stats, tibble, stringr, utils, recipes,
          multilevelmod, rules, poissonreg, censored, baguette, bonsai,
          brulee, rstanarm, dbarts, kknn, ranger, randomForest,
          LiblineaR, flexsurv, gee, glmnet, discrim, kernlab, klaR, mda,
          sda, sparsediscrim

VignetteBuilder  knitr

Imports  rlang (>= 0.4.11), purrr (>= 0.3.5), dplyr (>= 1.0.10),
         rsample (>= 1.1.0), workflows (>= 1.1.2), forcats,
         workflowsets, tidyr, broom, ggplot2, magrittr

RoxygenNote  7.3.1

NeedsCompilation  no

Author  Steven Sanderson [aut, cre, cph]

Maintainer  Steven Sanderson <spsanderson@gmail.com>

Repository  CRAN

Date/Publication  2024-03-12 14:10:02 UTC
core_packages

R topics documented:

- `core_packages` ................................................................. 2
- `create_model_spec` ........................................................... 3
- `create_splits` ................................................................. 4
- `create_workflow_set` ......................................................... 5
- `extract_model_spec` ......................................................... 7
- `extract_regression_residuals` .............................................. 8
- `extract_wflw` ................................................................. 9
- `extract_wflw_fit` ............................................................. 10
- `extract_wflw_pred` ............................................................ 11
- `fast_classification` .......................................................... 12
- `fast_classification_parsnip_spec_tbl` ................................... 13
- `fast_regression` .............................................................. 14
- `fast_regression_parsnip_spec_tbl` ....................................... 16
- `full_internal_make_wflw` .................................................. 17
- `get_model` ................................................................. 18
- `install_deps` ............................................................... 19
- `internal_make_fitted_wflw` ................................................. 20
- `internal_make_spec_tbl` ..................................................... 21
- `internal_make_wflw` .......................................................... 22
- `internal_make_wflw_gee_lin_reg` ......................................... 23
- `internal_make_wflw_predictions` ......................................... 24
- `internal_set_args_to_tune` ............................................... 25
- `load_deps` ................................................................. 26
- `make_classification_base_tbl` ............................................ 27
- `make_regression_base_tbl` ............................................... 28
- `match_args` ............................................................... 29
- `plot_regression_predictions` ............................................. 30
- `plot_regression_residuals` ................................................. 31

Index ................................................................................. 33

---

### Description

Lists the core packages necessary to run all potential modeling algorithms.

### Usage

```r
core_packages()
```

### Details

Lists the core packages necessary to run all potential modeling algorithms.
create_model_spec

Value

A character vector

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Utility: create_splits(), create_workflow_set(), fast_classification_parsnip_spec_tbl(), fast_regression_parsnip_spec_tbl(), full_internal_make_wflw(), install_deps(), load_deps(), match_args()

Examples

core_packages()

create_model_spec  Generate Model Specification calls to parsnip

Description

Creates a list/tibble of parsnip model specifications.

Usage

create_model_spec(
  .parsnip_eng = list("lm"),
  .mode = list("regression"),
  .parsnip_fns = list("linear_reg"),
  .return_tibble = TRUE
)

Arguments

  .parsnip_eng The input must be a list. The default for this is set to all. This means that all of the parsnip linear regression engines will be used, for example lm, or glm.
  .mode The input must be a list. The default is 'regression'
  .parsnip_fns The input must be a list. The default for this is set to all. This means that all of the parsnip linear regression functions will be used, for example linear_reg(), or cubist_rules.
  .return_tibble The default is TRUE. FALSE will return a list object.
create_splits

Utility Create Splits Object

Description

Create a splits object.

Usage

create_splits(.data, .split_type = "initial_split", .split_args = NULL)
create_workflow_set

Arguments

.data The data being passed to make a split on
.split_type The default is "initial_split", you can pass any other split type from the rsample library.
.split_args The default is NULL in order to use the default split arguments. If you want to pass other arguments then must pass a list with the parameter name and the argument.

Details

Create a splits object that returns a list object of both the splits object itself and the splits type. This function supports all splits types from the rsample package.

Value

A list object

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Utility: core_packages(), create_workflow_set(), fast_classification_parsnip_spec_tbl(), fast_regression_parsnip_spec_tbl(), full_internal_make_wflw(), install_deps(), load_deps(), match_args()

Examples

create_splits(mtcars, .split_type = "vfold_cv")
create_workflow_set

Arguments

- `.model_tbl` The model table that is generated from a function like `fast_regression_parsnip_spec_tbl()`. The model spec column will be grabbed automatically as the class of the object must be `tidyaml_base_tbl`
- `.recipe_list` Provide a list of recipes here that will get added to the workflow set object.
- `.cross` The default is `TRUE`, can be set to `FALSE`. This is passed to the `cross` parameter as an argument to the `workflow_set()` function.

Details

Create a workflow set object/tibble from a model spec tibble where the object class type is `tidyaml_base_tbl`. This function will take in a list of recipes and will grab the model specifications from the base tibble to create the workflow sets object. You can also supply the logical of `TRUE/FALSE` the `.cross` parameter which gets passed to the corresponding parameter as an argument to the `workflowsets::workflow_set()` function.

Value

A list object of workflows.

Author(s)

Steven P. Sanderson II, MPH

See Also

- [https://workflowsets.tidymodels.org/](https://workflowsets.tidymodels.org/)
- Other Utility: `core_packages()`, `create_splits()`, `fast_classification_parsnip_spec_tbl()`, `fast_regression_parsnip_spec_tbl()`, `full_internal_make_wflw()`, `install_deps()`, `load_deps()`, `match_args()`

Examples

```r
library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
spec_tbl <- fast_regression_parsnip_spec_tbl(
  .parsnip_fns = "linear_reg",
  .parsnip_eng = c("lm","glm")
)

create_workflow_set(
  spec_tbl,
  list(rec_obj)
)
```
extract_model_spec  

**Extract A Model Specification**

**Description**

Extract a model specification from a tidyAML model tibble.

**Usage**

```r
extract_model_spec(.data, .model_id = NULL)
```

**Arguments**

- `.data`: The model table that must have the class `tidyaml_mod_spec_tbl`.
- `.model_id`: The model number that you want to select. Must be an integer or sequence of integers, i.e. `1` or `c(1,3,5)` or `1:2`.

**Details**

This function allows you to get a model specification or more from a tibble with a class of "tidyaml_mod_spec_tbl". It allows you to select the model by the `.model_id` column. You can call the model id's by an integer or a sequence of integers.

**Value**

A tibble with the chosen model specification(s).

**Author(s)**

Steven P. Sanderson II, MPH

**See Also**

Other Extractor: `extract_regression_residuals()`, `extract_wflw()`, `extract_wflw_fit()`, `extract_wflw_pred()`, `get_model()`

**Examples**

```r
spec_tbl <- fast_regression_parsnip_spec_tbl(
  .parsnip_fns = "linear_reg",
  .parsnip_eng = c("lm","glm")
)

extract_model_spec(spec_tbl, 1)
extract_model_spec(spec_tbl, 1:2)
```
extract_regression_residuals

Extract Residuals from Fast Regression Models

Description
This function extracts residuals from a fast regression model table (fast_regression()).

Usage
extract_regression_residuals(.model_tbl, .pivot_long = FALSE)

Arguments
.model_tbl
A fast regression model specification table (fst_reg_spec_tbl).
.pivot_long
A logical value indicating if the output should be pivoted. The default is FALSE.

Details
The function checks if the input model specification table inherits the class 'fst_reg_spec_tbl' and if it contains the column 'pred_wflw'. It then manipulates the data, grouping it by model, and extracts residuals for each model. The result is a list of data frames, each containing residuals, actual values, and predicted values for a specific model.

Value
The function returns a list of data frames, each containing residuals, actual values, and predicted values for a specific model.

Author(s)
Steven P. Sanderson II, MPH

See Also
Other Extractor: extract_model_spec(), extract_wflw(), extract_wflw_fit(), extract_wflw_pred(), get_model()

Examples
library(recipes, quietly = TRUE)

rec_obj <- recipe(mpg ~ ., data = mtcars)

fr_tbl <- fast_regression(mtcars, rec_obj, .parsnip_eng = c("lm","glm"),
.parsnip_fns = "linear_reg")

extract_regression_residuals(fr_tbl)
extract_wflw

extract_regression_residuals(fr_tbl, .pivot_long = TRUE)

---

extract_wflw Extract A Model Workflow

Description

Extract a model workflow from a tidyAML model tibble.

Usage

extract_wflw(.data, .model_id = NULL)

Arguments

.data The model table that must have the class tidyaml_mod_spec_tbl.
.model_id The model number that you want to select. Must be an integer or sequence of integers, ie. 1 or c(1,3,5) or 1:2

Details

This function allows you to get a model workflow or more from a tibble with a class of "tidyaml_mod_spec_tbl". It allows you to select the model by the .model_id column. You can call the model id's by an integer or a sequence of integers.

Value

A tibble with the chosen model workflow(s).

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Extractor: extract_model_spec(), extract_regression_residuals(), extract_wflw_fit(), extract_wflw_pred(), get_model()

Examples

library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
frt_tbl <- fast_regression(mtcars, rec_obj, .parsnip_eng = c("lm","glm"),
                          .parsnip_fns = "linear_reg")

extract_wflw(frt_tbl, 1)
extract_wflw(frt_tbl, 1:2)
extract_wflw_fit

Extract A Model Fitted Workflow

Description

Extract a model fitted workflow from a tidyAML model tibble.

Usage

extract_wflw_fit(.data, .model_id = NULL)

Arguments

.data The model table that must have the class tidyaml_mod_spec_tbl.
.model_id The model number that you want to select. Must be an integer or sequence of integers, ie. 1 or c(1,3,5) or 1:2

Details

This function allows you to get a model fitted workflow or more from a tibble with a class of "tidyaml_mod_spec_tbl". It allows you to select the model by the .model_id column. You can call the model id's by an integer or a sequence of integers.

Value

A tibble with the chosen model workflow(s).

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Extractor: extract_model_spec(), extract_regression_residuals(), extract_wflw(), extract_wflw_pred(), get_model()

Examples

library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
frt_tbl <- fast_regression(mtcars, rec_obj, .parsnip_eng = c("lm","glm"),
                          .parsnip_fns = "linear_reg")

extract_wflw_fit(frt_tbl, 1)
extract_wflw_fit(frt_tbl, 1:2)
**extract_wflw_pred**

*Extract A Model Workflow Predictions*

**Description**

Extract a model workflow predictions from a tidyAML model tibble.

**Usage**

```r
extract_wflw_pred(.data, .model_id = NULL)
```

**Arguments**

- `.data` The model table that must have the class `tidyaml_mod_spec_tbl`.
- `.model_id` The model number that you want to select. Must be an integer or sequence of integers, i.e. 1 or c(1,3,5) or 1:2

**Details**

This function allows you to get a model workflow predictions or more from a tibble with a class of "tidyaml_mod_spec_tbl". It allows you to select the model by the `.model_id` column. You can call the model id's by an integer or a sequence of integers.

**Value**

A tibble with the chosen model workflow(s).

**Author(s)**

Steven P. Sanderson II, MPH

**See Also**

Other Extractor: `extract_model_spec()`, `extract_regression_residuals()`, `extract_wflw()`, `extract_wflw_fit()`, `get_model()`

**Examples**

```r
library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
frt_tbl <- fast_regression(mtcars, rec_obj, .parsnip_eng = c("lm","glm"),
                           .parsnip_fns = "linear_reg")

extract_wflw_pred(frt_tbl, 1)
extract_wflw_pred(frt_tbl, 1:2)
```
**fast_classification**  
*Generate Model Specification calls to parsnip*

**Description**

Creates a list/tibble of parsnip model specifications.

**Usage**

```r
fast_classification(
  .data,  
  .rec_obj,  
  .parsnip_fns = "all",  
  .parsnip_eng = "all",  
  .split_type = "initial_split",  
  .split_args = NULL,  
  .drop_na = TRUE  
)
```

**Arguments**

- `.data`: The data being passed to the function for the classification problem
- `.rec_obj`: The recipe object being passed.
- `.parsnip_fns`: The default is 'all' which will create all possible classification model specifications supported.
- `.parsnip_eng`: The default is 'all' which will create all possible classification model specifications supported.
- `.split_type`: The default is 'initial_split', you can pass any type of split supported by rsample.
- `.split_args`: The default is NULL, when NULL then the default parameters of the split type will be executed for the rsample split type.
- `.drop_na`: The default is TRUE, which will drop all NA's from the data.

**Details**

With this function you can generate a tibble output of any classification model specification and it's fitted workflow object. Per recipes documentation explicitly with step_string2factor() it is encouraged to mutate your predictor into a factor before you create your recipe.

**Value**

A list or a tibble.

**Author(s)**

Steven P. Sanderson II, MPH
See Also

Other Model Generator: `create_model_spec()`, `fast_regression()`

Examples

```r
library(recipes)
library(dplyr)
library(tidyr)

df <- Titanic |> as_tibble() |> uncount(n) |> mutate(across(everything(), as.factor))

rec_obj <- recipe(Survived ~ ., data = df)

fct_tbl <- fast_classification(
  .data = df,
  .rec_obj = rec_obj,
  .parsnip_eng = c("glm","earth")
)

fct_tbl
```

Description

Creates a tibble of parsnip classification model specifications.

Usage

```r
fast_classification_parsnip_spec_tbl(
  .parsnip_fns = "all",
  .parsnip_eng = "all"
)
```

Arguments

- `.parsnip_fns` The default for this is set to all. This means that all of the parsnip classification functions will be used, for example `bag_mars()`, or `bart()`. You can also choose to pass a c() vector like c("bag_mars", "bart")
- `.parsnip_eng` The default for this is set to all. This means that all of the parsnip classification engines will be used, for example `earth`, or `dbarts`. You can also choose to pass a c() vector like c(‘earth’, ’dbarts’)

Utility Classification call to parsnip
fast_regression

Description

Creates a list/tibble of parsnip model specifications.

Usage

```r
fast_regression(
  .data,
  .rec_obj,
  .parsnip_fns = "all",
  .parsnip_eng = "all",
  .split_type = "initial_split",
  .split_args = NULL,
  .drop_na = TRUE
)
```
**Arguments**

- `.data` The data being passed to the function for the regression problem
- `.rec_obj` The recipe object being passed.
- `.parsnip_fns` The default is 'all' which will create all possible regression model specifications supported.
- `.parsnip_eng` the default is 'all' which will create all possible regression model specifications supported.
- `.split_type` The default is 'initial_split', you can pass any type of split supported by `rsample`
- `.split_args` The default is NULL, when NULL then the default parameters of the split type will be executed for the `rsample` split type.
- `.drop_na` The default is TRUE, which will drop all NA's from the data.

**Details**

With this function you can generate a tibble output of any regression model specification and it's fitted workflow object.

**Value**

A list or a tibble.

**Author(s)**

Steven P. Sanderson II, MPH

**See Also**

Other Model_Generator: `create_model_spec()`, `fast_classification()`

**Examples**

```r
library(recipes, quietly = TRUE)

rec_obj <- recipe(mpg ~ ., data = mtcars)
frt_tbl <- fast_regression(
  mtcars, 
  rec_obj, 
  .parsnip_eng = c("lm","glm","gee"), 
  .parsnip_fns = "linear_reg"
)

frt_tbl
```
fast_regression_parsnip_spec_tbl

Utility Regression call to parsnip

Description

Creates a tibble of parsnip regression model specifications.

Usage

```
fast_regression_parsnip_spec_tbl(.parsnip_fns = "all", .parsnip_eng = "all")
```

Arguments

- `.parsnip_fns` The default for this is set to `all`. This means that all of the parsnip linear regression functions will be used, for example `linear_reg()`, or `cubist_rules`. You can also choose to pass a `c()` vector like `c("linear_reg","cubist_rules")`
- `.parsnip_eng` The default for this is set to `all`. This means that all of the parsnip linear regression engines will be used, for example `lm`, or `glm`. You can also choose to pass a `c()` vector like `c('lm', 'glm')`

Details

Creates a tibble of parsnip regression model specifications. This will create a tibble of 46 different regression model specifications which can be filtered. The model specs are created first and then filtered out. This will only create models for regression problems. To find all of the supported models in this package you can visit [https://www.tidymodels.org/find/parsnip/](https://www.tidymodels.org/find/parsnip/)

Value

A tibble with an added class of `fst_reg_spec_tbl`

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Utility: `core_packages()`, `create_splits()`, `create_workflow_set()`, `fast_classification_parsnip_spec_tbl()`, `full_internal_make_wflw()`, `install_deps()`, `load_deps()`, `match_args()

Examples

```
fast_regression_parsnip_spec_tbl(.parsnip_fns = "linear_reg")
fast_regression_parsnip_spec_tbl(.parsnip_eng = c("lm","glm"))
```
full_internal_make_wflw

Full Internal Workflow for Model and Recipe

Description
This function creates a full internal workflow for a model and recipe combination.

Usage
full_internal_make_wflw(.model_tbl, .rec_obj)

Arguments
- .model_tbl: A model specification table (tidyml_mod_spec_tbl).
- .rec_obj: A recipe object.

Details
The function checks if the input model specification table inherits the class 'tidyml_mod_spec_tbl'. It then manipulates the input table, making adjustments for factors and creating a list of grouped models. For each model-recipe pair, it uses the appropriate internal function based on the model type to create a workflow object. The specific internal function is selected using a switch statement based on the class of the model.

Value
The function returns a workflow object for the first model-recipe pair based on the internal function selected.

Author(s)
Steven P. Sanderson II, MPH

See Also
Other Utility: core_packages(), create_splits(), create_workflow_set(), fast_classification_parsnip_spec_tbl(), fast_regression_parsnip_spec_tbl(), install_deps(), load_deps(), match_args()

Examples
library(dplyr)
library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
mod_tbl <- make_regression_base_tbl()
mod_tbl <- mod_tbl |>
get_model

Get a Model

Description

Get a model from a tidyAML model tibble.

Usage

get_model(.data, .model_id = NULL)

Arguments

.data The model table that must have the class tidyaml_mod_spec_tbl.
.model_id The model number that you want to select. Must be an integer or sequence of integers, i.e. 1 or c(1,3,5) or 1:2

Details

This function allows you to get a model or models from a tibble with a class of "tidyaml_mod_spec_tbl". It allows you to select the model by the .model_id column. You can call the model id's by an integer or a sequence of integers.

Value

A tibble with the chosen models.

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Extractor: extract_model_spec(), extract_regression_residuals(), extract_wflw(), extract_wflw_fit(), extract_wflw_pred()
install_deps

Examples

library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
spec_tbl <- fast_regression_parsnip_spec_tbl(
  .parsnip_fns = "linear_reg",
  .parsnip_eng = c("lm","glm")
)

get_model(spec_tbl, 1)
get_model(spec_tbl, 1:2)

install_deps

Functions to Install all Core Libraries

Description

Installs all dependencies in the core_packages() function.

Usage

install_deps()

Details

Installs all dependencies in the core_packages() function.

Value

No return value, called for side effects

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Utility: core_packages(), create_splits(), create_workflow_set(), fast_classification_parsnip_spec_tbl(),
fast_regression_parsnip_spec_tbl(), full_internal_make_wflw(), load_deps(), match_args()

Examples

## Not run:
install_deps()

## End(Not run)
Internal_make_fitted_wflw

Internals Safely Make a Fitted Workflow from Model Spec tibble

Description

Safely Make a fitted workflow from a model spec tibble.

Usage

internal_make_fitted_wflw(.model_tbl, .splits_obj)

Arguments

.model_tbl
   The model table that is generated from a function like fast_regression_parsnip_spec_tbl(),
   must have a class of "tidy YAML_mod_spec_tbl". This is meant to be used after the
   function internal_make_wflw() has been run and the tibble has been saved.
.splits_obj
   The splits object from the auto_ml function. It is internal to the auto_ml function.

Details

Create a fitted parsnip model from a workflow object.

Value

A list object of workflows.

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Internals: internal_make_spec_tbl(), internal_make_wflw(), internal_make_wflw_gee_lin_reg(),
 internal_make_wflw_predictions(), internal_set_args_to_tune(), make_classification_base_tbl(),
 make_regression_base_tbl()

Examples

library(recipes, quietly = TRUE)

mod_spec_tbl <- fast_regression_parsnip_spec_tbl(
  .parsnip_eng = c("lm", "glm"),
  .parsnip_fns = "linear_reg"
)

rec_obj <- recipe(mpg ~ ., data = mtcars)
splits_obj <- create_splits(mtcars, "initial_split")
mod_tbl <- mod_spec_tbl |>
  mutate(wflw = full_internal_make_wflw(mod_spec_tbl, rec_obj))
internal_make_fitted_wflw(mod_tbl, splits_obj)

internal_make_spec_tbl

Internals Make a Model Spec tibble

Description
Make a Model Spec tibble.

Usage
internal_make_spec_tbl(.model_tbl)

Arguments
.model_tbl This is the data that should be coming from inside of the regression/classification
to parsnip spec functions.

Details
Make a Model Spec tibble.

Value
A model spec tbl.

Author(s)
Steven P. Sanderson II, MPH

See Also
Other Internals: internal_make_fitted_wflw(), internal_make_wflw(), internal_make_wflw_gee_lin_reg(),
internal_make_wflw_predictions(), internal_set_args_to_tune(), make_classification_base_tbl(),
make_regression_base_tbl()
Examples

```r
make_regression_base_tbl() |> internal_make_spec_tbl()
make_classification_base_tbl() |> internal_make_spec_tbl()
```

---

**internal_make_wflw**  
*Internals Safely Make Workflow from Model Spec tibble*

**Description**

Safely Make a workflow from a model spec tibble.

**Usage**

```r
internal_make_wflw(.model_tbl, .rec_obj)
```

**Arguments**

- `.model_tbl`: The model table that is generated from a function like `fast_regression_parsnip_spec_tbl()`, must have a class of "tidyml_mod_spec_tbl".
- `.rec_obj`: The recipe object that is going to be used to make the workflow object.

**Details**

Create a model specification tibble that has a `workflows::workflow()` list column.

**Value**

A list object of workflows.

**Author(s)**

Steven P. Sanderson II, MPH

**See Also**

Other Internals: `internal_make_fitted_wflw()`, `internal_make_spec_tbl()`, `internal_make_wflw_glm()`, `internal_make_wflw_predictions()`, `internal_set_args_to_tune()`, `make_classification_base_tbl()`, `make_regression_base_tbl()`
### Examples

```r
library(recipes, quietly = TRUE)

mod_spec_tbl <- fast_regression_parsnip_spec_tbl(
  .parsnip_eng = c("lm", "glm", "gee"),
  .parsnip_fns = "linear_reg"
)

rec_obj <- recipe(mpg ~ ., data = mtcars)

internal_make_wflw(mod_spec_tbl, rec_obj)
```

---

### Description

Safely Make a workflow from a model spec tibble.

### Usage

```r
internal_make_wflw_gee_lin_reg(.model_tbl, .rec_obj)
```

### Arguments

- `model_tbl` The model table that is generated from a function like `fast_regression_parsnip_spec_tbl()`, must have a class of "tidyml_mod_spec_tbl".
- `rec_obj` The recipe object that is going to be used to make the workflow object.

### Details

Create a model specification tibble that has a `workflows::workflow()` list column.

### Value

A list object of workflows.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

Other Internals: `internal_make_fitted_wflw()`, `internal_make_spec_tbl()`, `internal_make_wflw()`, `internal_make_wflw_predictions()`, `internal_set_args_to_tune()`, `make_classification_base_tbl()`, `make_regression_base_tbl()`
internal_make_wflw_predictions

Internals Safely Make Predictions on a Fitted Workflow from Model Spec tibble

Description
Safely Make predictions on a fitted workflow from a model spec tibble.

Usage
internal_make_wflw_predictions(.model_tbl, .splits_obj)

Arguments
.model_tbl The model table that is generated from a function like fast_regression_parsnip_spec_tbl(), must have a class of "tidyml_mod_spec_tbl". This is meant to be used after the function internal_make_fitted_wflw() has been run and the tibble has been saved.
.splits_obj The splits object from the auto_ml function. It is internal to the auto_ml function.

Details
Create predictions on a fitted parsnip model from a workflow object.

Value
A list object tibble of the outcome variable and it’s values along with the testing and training predictions in a single tibble.
### Description

Make a tuned model specification object.
Usage

```
internal_set_args_to_tune(.model_tbl)
```

Arguments

- `.model_tbl` The model table that is generated from a function like `fast_regression_parsnip_spec_tbl()`, must have a class of "tidyml_mod_spec_tbl".

Details

This will take a model specification that is created from a function like `fast_regression_parsnip_spec_tbl()` and update the `model_spec` args to `tune::tune()`. This is done dynamically, meaning you do not need to know the names of the parameters inside of the model specification.

Value

A list object of workflows.

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Internals: `internal_make_fitted_wflw()`, `internal_make_spec_tbl()`, `internal_make_wflw()`, `internal_make_wflw_gee_lin_reg()`, `internal_make_wflw_predictions()`, `make_classification_base_tbl()`, `make_regression_base_tbl()`

Examples

```
library(dplyr)

mod_tbl <- fast_regression_parsnip_spec_tbl()
mod_tbl$model_spec[[1]]

updated_mod_tbl <- mod_tbl |
  mutate(model_spec = internal_set_args_to_tune(mod_tbl))
updated_mod_tbl$model_spec[[1]]
```

load_deps

Functions to Install all Core Libraries

Description

Load all the core packages necessary to run all potential modeling algorithms.
Usage

load_deps()

Details

Load all the core packages necessary to run all potential modeling algorithms.

Value

No return value, called for side effects

Author(s)

Steven P. Sanderson II, MPH

See Also

Other Utility: core_packages(), create_splits(), create_workflow_set(), fast_classification_parsnip_spec_tbl(), fast_regression_parsnip_spec_tbl(), full_internal_make_wflw(), install_deps(), match_args()

Examples

## Not run:
load_deps()

## End(Not run)
Author(s)
Steven P. Sanderson II, MPH

See Also
Other Internals: `internal_make_fitted_wflw()`, `internal_make_spec_tbl()`, `internal_make_wflw()`, `internal_make_wflw_gee_lin_reg()`, `internal_make_wflw_predictions()`, `internal_set_args_to_tune()`, `make_regression_base_tbl()

Examples
`make_classification_base_tbl()`

Description
Creates a base tibble to create parsnip regression model specifications.

Usage
`make_regression_base_tbl()`

Details
Creates a base tibble to create parsnip regression model specifications.

Value
A tibble

Author(s)
Steven P. Sanderson II, MPH

See Also
Other Internals: `internal_make_fitted_wflw()`, `internal_make_spec_tbl()`, `internal_make_wflw()`, `internal_make_wflw_gee_lin_reg()`, `internal_make_wflw_predictions()`, `internal_set_args_to_tune()`, `make_classification_base_tbl()

Examples
`make_regression_base_tbl()`
**match_args**

**Match function arguments**

**Description**

Match a functions arguments.

**Usage**

`match_args(f, args)`

**Arguments**

- **f**
  - The parsnip function such as "linear_reg" as a string and without the parentheses.
- **args**
  - The arguments you want to supply to f

**Details**

Match a functions arguments, the bad ones passed will be rejected but the remaining passing ones will be returned.

**Value**

A list of matched arguments.

**Author(s)**

Steven P. Sanderson II, MPH

**See Also**

Other Utility: `core_packages()`, `create_splits()`, `create_workflow_set()`, `fast_classification_parsnip_spec_tbl()`, `fast_regression_parsnip_spec_tbl()`, `full_internal_make_wflw()`, `install_deps()`, `load_deps()`

**Examples**

```r
match_args(
  f = "linear_reg",
  args = list(
    mode = "regression",
    engine = "lm",
    trees = 1,
    mtry = 1
  )
)
```
Description
Create a ggplot2 plot of regression predictions.

Usage
plot_regression_predictions(.data, .output = "list")

Arguments
.data The data from the output of the extract_regression_residuals() function.
.output The default is "list" which will return a list of plots. The other option is "facet" which will return a single faceted plot.

Details
Create a ggplot2 plot of regression predictions, the actual, training, and testing values. The output of this function can either be a list of plots or a single faceted plot. This function takes the output of the function extract_wflw_pred() function.

Value
A list of ggplot2 plots or a faceted plot.

Author(s)
Steven P. Sanderson II, MPH

See Also
Other Plotting: plot_regression_residuals()

Examples
library(recipes)
rec_obj <- recipe(mpg ~ ., data = mtcars)
frt_tbl <- fast_regression(mtcars, rec_obj, .parsnip_eng = c("lm","glm"), .parsnip_fns = "linear_reg")
plot_regression_residuals

Create ggplot2 plot of regression residuals

Description
Create a ggplot2 plot of regression residuals.

Usage
plot_regression_residuals(.data)

Arguments
.data
The data from the output of the extract_regression_residuals() function.

Details
Create a ggplot2 plot of regression residuals. The output of this function can either be a list of plots or a single faceted plot. This function takes the output of the extract_regression_residuals() function.

Value
A list of ggplot2 plots or a faceted plot.

Author(s)
Steven P. Sanderson II, MPH

See Also
Other Plotting: plot_regression_predictions()

Examples
library(recipes)

rec_obj <- recipe(mpg ~ ., data = mtcars)
frt_tbl <- fast_regression(
    mtcars,
    rec_obj,
    .parsnip_eng = c("lm","glm"),
    .parsnip_fns = "linear_reg"
)
plot_regression_residuals

)

extract_regression_residuals(frt_tbl, FALSE)[1] |> plot_regression_residuals()
extract_regression_residuals(frt_tbl, TRUE)[1] |> plot_regression_residuals()
Index

* **Extractor**
  extract_model_spec, 7
  extract_regression_residuals, 8
  extract_wflw, 9
  extract_wflw_fit, 10
  extract_wflw_pred, 11
  get_model, 18

* **Internals**
  internal_make_fitted_wflw, 20
  internal_make_spec_tbl, 21
  internal_make_wflw, 22
  internal_make_wflw_glm_reg, 23
  internal_make_wflw_predictions, 24
  internal_set_args_to_tune, 25
  make_classification_base_tbl, 27
  make_regression_base_tbl, 28

* **Model_Generator**
  create_model_spec, 3
  fast_classification, 12
  fast_regression, 14

* **Plotting**
  plot_regression_predictions, 30
  plot_regression_residuals, 31

* **Utility**
  core_packages, 2
  create_splits, 4
  create_workflow_set, 5
  fast_classification_parsnip_spec_tbl, 13
  fast_regression_parsnip_spec_tbl, 16
  full_internal_make_wflw, 17
  install_deps, 19
  load_deps, 26
  match_args, 29

  core_packages, 2, 5, 6, 14, 16, 17, 19, 27, 29
  create_model_spec, 3, 13, 15
  create_splits, 3, 4, 6, 14, 16, 17, 19, 27, 29
  create_workflow_set, 3, 5, 5, 14, 16, 17, 19, 27, 29
  extract_model_spec, 7, 8–11, 18
  extract_regression_residuals, 7, 8, 9–11, 18
  extract_wflw, 7, 8, 9, 10, 11, 18
  extract_wflw_fit, 7–9, 10, 11, 18
  extract_wflw_pred, 7–10, 11, 18
  fast_classification, 4, 12, 15
  fast_classification_parsnip_spec_tbl, 3, 5, 6, 13, 16, 17, 19, 27, 29
  fast_regression, 4, 13, 14
  fast_regression_parsnip_spec_tbl, 3, 5, 6, 14, 16, 17, 19, 27, 29
  get_model, 7–11, 18
  install_deps, 3, 5, 6, 14, 16, 17, 19, 27, 29
  internal_make_fitted_wflw, 20, 21–23, 25, 26, 28
  internal_make_spec_tbl, 20, 21, 22, 23, 25, 26, 28
  internal_make_wflw, 20, 21, 22, 23, 25, 26, 28
  internal_make_wflw_glm_reg, 20–22, 23, 25, 26, 28
  internal_make_wflw_predictions, 20–23, 24, 26, 28
  internal_set_args_to_tune, 20–23, 25, 28
  load_deps, 3, 5, 6, 14, 16, 17, 19, 26, 29
  make_classification_base_tbl, 20–23, 25, 26, 27, 28

33
make_regression_base_tbl, 20–23, 25, 26, 28, 28
match_args, 3, 5, 6, 14, 16, 17, 19, 27, 29
plot_regression_predictions, 30, 31
plot_regression_residuals, 30, 31
workflows::workflow(), 22, 23
workflowsets::workflow_set(), 6