Package: modelgrid (via r-universe)

November 15, 2026

Title A Framework for Creating, Managing and Training Multiple 'caret' Models

Version 1.2.0.9001

Description A minimalistic but flexible framework that facilitates the creation, management and training of multiple 'caret' models. A model grid consists of two components, (1) a set of settings that is shared by all models by default, and (2) specifications that apply only to the individual models. When the model grid is trained, model and training specifications are first consolidated from the shared and the model specific settings into complete 'caret' model configurations. These models are then trained with the 'train()' function from the 'caret' package. Compatibility updates by Renova.

URL https://github.com/smaakage85/modelgrid

Depends R (>= 3.4.0)

License MIT + file LICENSE

Encoding UTF-8

Imports caret, purrr, dplyr, magrittr

RoxygenNote 7.2.3

Suggests testthat, knitr, rmarkdown, recipes

VignetteBuilder knitr

NeedsCompilation no

Author Lars Kjeldgaard [aut], Angel Robles [cre] (<https://orcid.org/0000-0002-4674-4270>)

Maintainer Angel Robles <a.l.robles.fernandez@gmail.com>

Config/pak/sysreqs libicu-dev

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-15 20:26:38 UTC

RemoteUrl https://github.com/renova-r/modelgrid

RemoteRef renova-1.2.0.9001

RemoteSha 307b9cb28dbf998de91342c07d34baa79e20eb0f

Contents

add_model 2
consolidate_model 3
edit_model 4
model_grid 4
remove_model 6
share_settings 7

Index **9**

`add_model` *Add a model specification to a model grid*

Description

Define and add an individual model (and model training) specification to an existing model grid.

Usage

```
add_model(model_grid, model_name = NULL, custom_control = NULL, ...)
```

Arguments

`model_grid`	`model_grid`
`model_name`	`character`, your custom name for a given model. Must be unique within the model grid. If you do not provide a name, the model will be given a generic name - 'Model[int]'.
`custom_control`	`list`, any customization to subsettings of the 'trControl' component from the 'shared_settings' of the model grid (will only work if trControl' parameter has actually been set as part of the shared settings).
`...`	All (optional) individual settings (including model training settings) that the user wishes to set for the new model.

Value

`model_grid` with an additional individual model specification.

Examples

```
library(magrittr)

# Pre-allocate empty model grid.
mg <- model_grid()

# Add 'random forest' model spec.
mg <-
  mg %>%
  add_model(model_name = "Random Forest Test", method = "rf", tuneLength = 5)
```

consolidate_model *Consolidate model settings to a complete caret model specification*

Description

Consolidate model (and model training) settings from shared and model specific settings to one complete caret model specification. In case there is an overlap between the two, the model specific settings will apply.

Usage

```
consolidate_model(shared_settings, model)
```

Arguments

shared_settings	list, settings that are shared by all models by default.
model	list, the individual specifications of a model in a model grid.

Value

list, a complete model and training specification, that can be trained with caret.

Examples

```
library(magrittr)
library(dplyr)
library(caret)

# create model grid.
mg <-
  model_grid() %>%
  share_settings(y = iris[["Species"]],
                 x = iris %>% select(-Species),
                 trControl = trainControl()) %>%
  add_model("FunkyForest", method = "rf",
            preProc = c("center", "scale", "pca"),
            custom_control = list(preProcOptions = list(thresh = 0.8)))

# consolidate all settings to complete caret model specification.
consolidate_model(mg$shared_settings, mg$models$FunkyForest)
```

edit_model *Edit model within a model grid*

Description

Modify an existing model (and training) specification in a model grid.

Usage

```
edit_model(model_grid, model_name, ...)
```

Arguments

model_grid	model_grid
model_name	character, the unique name (as set by the user) of the model, that should be modified.
...	All the model (and model training) settings you want to modify for an existing model specification.

Value

model_grid

Examples

```
library(magrittr)

# Create model grid and add random forest model.
mg <-
  model_grid() %>%
  add_model(model_name = "Random Forest Test", method = "rf", tuneLength = 5)

# Edit the size of tuning grid of the random forest model.
edit_model(mg, model_name = "Random Forest Test", tuneLength = 10)
```

model_grid *Pre-allocate an empty model grid*

Description

Constructor function that pre-allocates an empty model grid. The model grid makes it easy to create, manage and train multiple caret models. Define the settings that by default are to be shared by all of the models in the model grid with share_settings. Add the individual specifications for the models you want to investigate with add_model. Train all of the models in the model grid with train.

The S3 method of the train function for the 'model_grid' class consolidates all model (and training) configurations from a model grid and trains them with the train function from the caret package.

Usage

```
model_grid()

## S3 method for class 'model_grid'
train(x, ..., train_all = FALSE, resample_seed = 123)
```

Arguments

x	model_grid
...	other arguments passed to methods across models in order to obtain a fair (and reproducible) comparison of the models. If set to NULL, seed will not be set (NOT advised).
train_all	logical if set to TRUE, all models will be trained. If set to FALSE, only models, for which no fit already exists, will be trained.
resample_seed	integer is used to create identical resamples

Value

model_grid

model_grid equipped with fitted models.

See Also

add_model for how to add a model to a model grid, edit_model for how to edit an existing model within a model grid, share_settings for how to define the shared settings of models within a model grid, consolidate_model for how to consolidate the shared settings of a model grid and the individual settings of a given model into one complete caret model configuration and remove_model for how to remove a model from a model grid.

Examples

```
# Pre-allocate an empty model grid.
model_grid()


library(caret)
library(magrittr)
library(dplyr)
data(GermanCredit)

# Create model grid with two different Random Forest models.
mg <-
  model_grid() %>%
  share_settings(
    y = GermanCredit[["Class"]],
    x = GermanCredit %>% select(-Class),
    metric = "ROC",
    trControl = trainControl(
      method = "cv",
```

```
    number = 2,
    summaryFunction = twoClassSummary,
    classProbs = TRUE
  )
) %>%
add_model(
  model_name = "RF",
  method = "rf",
  tuneLength = 3
  ) %>%
add_model(
  model_name = "RF NZV",
  method = "rf",
  preProc = "nzv",
  tuneGrid = data.frame(mtry = c(2, 10))
  )

# Train all model configurations in model grid.
train(mg)
```

`remove_model` *Remove model from model grid*

Description

Removes an individual model specification from a model grid. If the model has been trained, the fitted model will also be deleted.

Usage

```
remove_model(model_grid, model_name)
```

Arguments

`model_grid`	`model_grid`
`model_name`	`character`, the unique name (as set by the user) of the model, which will be removed from a model grid.

Value

`model_grid`

Examples

```
library(magrittr)

# Pre-allocate empty model grid.
mg <- model_grid()
```

```
# Add random forest model.
mg <-
  mg %>%
  add_model(model_name = "Random Forest Test", method = "rf", tuneLength = 5)

# Remove random forest model again.
remove_model(mg, model_name = "Random Forest Test")
```

`share_settings` *Set shared settings of a model grid*

Description

Set shared settings for all model (and training) configurations within a model grid. These settings will apply for any given model, unless the same settings have already been specified in the model specific configurations. In that case, the model specific settings will apply.

Usage

```
share_settings(model_grid, ...)
```

Arguments

`model_grid`	`model_grid`
`...`	All optional shared settings.

Value

`model_grid` equipped with shared settings.

Examples

```
library(magrittr)
library(caret)
library(dplyr)
data(GermanCredit)

# Pre-allocate empty model grid.
models <- model_grid()

# Set shared settings of model grid.
models %>%
  share_settings(
    y = GermanCredit[["Class"]],
    x = GermanCredit %>% select(-Class),
    metric = "ROC",
    preProc = c("center", "scale", "pca"),
    trControl = trainControl(
```

```
    method = "cv",
    number = 5,
    summaryFunction = twoClassSummary,
    classProbs = TRUE
    )
)
```

Index

add_model, 2, *4, 5*

consolidate_model, 3, *5*

edit_model, 4, *5*

model_grid, 4

remove_model, *5*, 6

share_settings, *4, 5*, 7

train, *4*
train.model_grid *(*model_grid*)*, 4