Package ‘modeltime.resample’

April 12, 2023

Title  Resampling Tools for Time Series Forecasting

Version  0.2.3

Description  A 'modeltime' extension that implements forecast resampling tools that assess time-based model performance and stability for a single time series, panel data, and cross-sectional time series analysis.

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  true

URL  https://github.com/business-science/modeltime.resample

BugReports  https://github.com/business-science/modeltime.resample/issues

Depends  modeltime (>= 0.3.0), R (>= 3.5)

Imports  tune, rsample, workflows, parsnip (>= 0.1.4), recipes, dials, yardstick, timetk (>= 2.5.0), tibble, dplyr, tidyr, purrr, forcats, glue, stringr, ggrepplot2, plotly, cli, crayon, magrittr, rlang (>= 0.1.2), progressr, tictoc, hardhat

Suggests  roxygen2, testthat, tidymodels, tidyverse, tidyquant, glmnet, lubridate, knitr, rmarkdown, covr, remotes

RoxygenNote  7.2.3

VignetteBuilder  knitr

NeedsCompilation  no

Author  Matt Dancho [aut, cre], Business Science [cph]

Maintainer  Matt Dancho <mdancho@business-science.io>

Repository  CRAN

Date/Publication  2023-04-12 15:50:02 UTC
**get_target_text_from_resamples**

### Description

An internal function used by `unnest_modeltime_resamples()`.

### Usage

```r
get_target_text_from_resamples(data, column_before_target = ".row")
```

### Arguments

- **data**: Unnested resample results
- **column_before_target**: The text column located before the target variable. This is ".row".

### Examples

```r
# The .resample_results column is deeply nested
m750_training_resamples_fitted

# Unnest and prepare the resample predictions for evaluation
unnest_modeltime_resamples(m750_training_resamples_fitted) %>%
  get_target_text_from_resamples()
```
m750_training_resamples_fitted

Time Series Cross Validation Resample Predictions (Results) from the M750 Data (Training Set)

Description

Time Series Cross Validation Resample Predictions (Results) from the M750 Data (Training Set)

Usage

m750_training_resamples_fitted

Format

A Modeltime Table that has been fitted to resamples with predictions in the .resample_results column

Details

m750_training_resamples_fitted <- m750_models %>%
  modeltime_fit_resamples(
    resamples = m750_training_resamples,
    control = control_resamples(verbose = T)
  )

See Also

- modeltime::m750_models
- modeltime::m750_training_resamples

Examples

m750_training_resamples_fitted
Description
Resampled predictions are commonly used for:

1. Analyzing accuracy and stability of models
2. As inputs to Ensemble methods (refer to the modeltime.ensemble package)

Usage
modeltime_fit_resamples(object, resamples, control = control_resamples())

Arguments
object A Modeltime Table
resamples An rset resample object. Used to generate sub-model predictions for the meta-learner. See timetk::time_series_cv() or rsample::vfold_cv() for making resamples.
control A tune::control_resamples() object to provide control over the resampling process.

Details
The function is a wrapper for tune::fit_resamples() to iteratively train and predict models contained in a Modeltime Table on resample objects. One difference between tune::fit_resamples() and modeltime_fit_resamples() is that predictions are always returned (i.e. control = tune::control_resamples(save = TRUE)). This is needed for ensemble_model_spec().

Resampled Prediction Accuracy
Calculating Accuracy Metrics on models fit to resamples can help to understand the model performance and stability under different forecasting windows. See modeltime_resample_accuracy() for getting resampled prediction accuracy for each model.

Ensembles
Fitting and Predicting Resamples is useful in creating Stacked Ensembles using modeltime.ensemble::ensemble_model_spec(). The sub-model cross-validation predictions are used as the input to the meta-learner model.

Value
A Modeltime Table (mdl_time_tbl) object with a column containing resample results (.resample_results)
Examples

```r
library(tidymodels)
library(modeltime)
library(timetk)
library(tidyverse)

# Make resamples
resamples_tscv <- training(m750_splits) %>%
  time_series_cv(
    assess = "2 years",
    initial = "5 years",
    skip = "2 years",
    # Normally we do more than one slice, but this speeds up the example
    slice_limit = 1
  )

# Fit and generate resample predictions
m750_models_resample <- m750_models %>%
  modeltime_fit_resamples(
    resamples = resamples_tscv,
    control = control_resamples(verbos = TRUE)
  )

# A new data frame is created from the Modeltime Table
# with a column labeled, '
  m750_models_resample
```

---

**modeltime_resample_accuracy**

*Calculate Accuracy Metrics from Modeltime Resamples*

**Description**

This is a wrapper for yardstick that simplifies time series regression accuracy metric calculations from a Modeltime Table that has been resampled and fitted using `modeltime_fit_resamples()`.

**Usage**

```r
modeltime_resample_accuracy(
  object,
  summary_fns = mean,
  metric_set = default_forecast_accuracy_metric_set(),
  ...
)
```
Arguments

object a Modeltime Table with a column '.resample_results' (the output of `modeltime_fit_resamples()`)  
summary_fns One or more functions to analyze resamples. The default is `mean()`. Possible values are:

- NULL, to returns the resamples untransformed.
- A function, e.g. `mean`.
- A purrr-style lambda, e.g. `~ mean(x, na.rm = TRUE)`
- A list of functions/lambdas, e.g. `list(mean = mean, sd = sd)`

metric_set A `yardstick::metric_set()` that is used to summarize one or more forecast accuracy (regression) metrics.

... Additional arguments passed to the function calls in `summary_fns`.

Details

#` Default Accuracy Metrics

The following accuracy metrics are included by default via `modeltime::default_forecast_accuracy_metric_set()`:

- MAE - Mean absolute error, `yardstick::mae()`
- MAPE - Mean absolute percentage error, `yardstick::mape()`
- MASE - Mean absolute scaled error, `yardstick::mase()`
- SMAPE - Symmetric mean absolute percentage error, `yardstick::smape()`
- RMSE - Root mean squared error, `yardstick::rmse()`
- RSQ - R-squared, `yardstick::rsq()`

Summary Functions

By default, `modeltime_resample_accuracy()` returns the average accuracy metrics for each resample prediction.

The user can change this default behavior using `summary_fns`. Simply pass one or more Summary Functions. Internally, the functions are passed to `dplyr::across(.fns)`, which applies the summary functions.

Returning Unsummarized Results

You can pass `summary_fns = NULL` to return unsummarized results by `.resample_id`.

Professional Tables (Interactive & Static)

Use `modeltime::table_modeltime_accuracy()` to format the results for reporting in reactable (interactive) or `gt` (static) formats, which are perfect for Shiny Apps (interactive) and PDF Reports (static).

Examples

```r
library(modeltime)

# Mean (Default)
m750_training_resamples_fitted %>%
modeltime_resample_accuracy() %>%
```
# Mean and Standard Deviation

```r
m750_training_resamples_fitted %>%
  modetime_resample_accuracy(
    summary_fns = list(mean = mean, sd = sd)
  )
```

# When summary_fns = NULL, returns the unsummarized resample results

```r
m750_training_resamples_fitted %>%
  modetime_resample_accuracy(
    summary_fns = NULL
  )
```

## Description

A convenient plotting function for visualizing resampling accuracy by resample set for each model in a Modeltime Table.

## Usage

```r
plot_modeltime_resamples(
  .data,
  .metric_set = default_forecast_accuracy_metric_set(),
  .summary_fn = mean,
  ..., 
  .facet_ncol = NULL,
  .facet_scales = "free_x",
  .point_show = TRUE,
  .point_size = 1,
  .point_shape = 16,
  .point_alpha = 1,
  .summary_line_show = TRUE,
  .summary_line_size = 0.5,
  .summary_line_type = 1,
  .summary_line_alpha = 1,
  .x_intercept = NULL,
  .x_intercept_color = "red",
  .x_intercept_size = 0.5,
  .legend_show = TRUE,
  .legend_max_width = 40,
  .title = "Resample Accuracy Plot",
)```
Arguments

.data A modeltime table that includes a column .resample_results containing the resample results. See modeltime_fit_resamples() for more information.

.metric_set A yardstick::metric_set() that is used to summarize one or more forecast accuracy (regression) metrics.

.summary_fn A single summary function that is applied to aggregate the metrics across resample sets. Default: mean.

... Additional arguments passed to the .summary_fn.

.facet_ncol Default: NULL. The number of facet columns.

.facet_scales Default: free_x.

.point_show Whether or not to show the individual points for each combination of models and metrics. Default: TRUE.

.point_size Controls the point size. Default: 1.

.point_shape Controls the point shape. Default: 16.

.point_alpha Controls the opacity of the points. Default: 1 (full opacity).

.summary_line_show Whether or not to show the summary lines. Default: TRUE.

.summary_line_size Controls the summary line size. Default: 0.5.

.summary_line_type Controls the summary line type. Default: 1.

.summary_line_alpha Controls the summary line opacity. Default: 1 (full opacity).

.x_intercept Numeric. Adds an x-intercept at a location (e.g. 0). Default: NULL.

.x_intercept_color Controls the x-intercept color. Default: "red".

.x_intercept_size Controls the x-intercept size. Default: 0.5.

.legend_show Logical. Whether or not to show the legend. Can save space with long model descriptions.

.legend_max_width Numeric. The width of truncation to apply to the legend text.

.title Title for the plot.

.x_lab X-axis label for the plot

.y_lab Y-axis label for the plot

.color_lab Legend label if a color_var is used.

.interactive Returns either a static (ggplot2) visualization or an interactive (plotly) visualization.
Details

Default Accuracy Metrics

The following accuracy metrics are included by default via `modeltime::default_forecast_accuracy_metric_set()`:  

- **MAE** - Mean absolute error, `yardstick::mae()`  
- **MAPE** - Mean absolute percentage error, `yardstick::mape()`  
- **MASE** - Mean absolute scaled error, `yardstick::mase()`  
- **SMAPE** - Symmetric mean absolute percentage error, `yardstick::smape()`  
- **RMSE** - Root mean squared error, `yardstick::rmse()`  
- **RSQ** - R-squared, `yardstick::rsq()`  

Summary Function

Users can supply a single summary function (e.g. `mean`) to summarize the resample metrics by each model.

Examples

```r
m750_training_resamples_fitted %>%
  plot_modeltime_resamples(.interactive = FALSE)
```

unnest_modeltime_resamples

*Unnest the Results of Modeltime Fit Resamples*

Description

An internal function used by `modeltime_resample_accuracy()`.

Usage

```
unnest_modeltime_resamples(object)
```

Arguments

- `object` A Modeltime Table that has a column `.resample_results`
Details

The following data columns are unnested and prepared for evaluation:

- `.row_id` - A unique identifier to compare observations.
- `.resample_id` - A unique identifier given to the resample iteration.
- `.model_id` and `.model_desc` - Modeltime Model ID and Description
- `.pred` - The Resample Prediction Value
- `.row` - The actual row value from the original dataset
- *Actual Value Column* - The name changes to target variable name in dataset

Value

Tibble with columns for `.row_id`, `.resample_id`, `.model_id`, `.model_desc`, `.pred`, `.row`, and actual value name from the data set

Examples

```r
# The .resample_results column is deeply nested
m750_training_resamples_fitted

# Unnest and prepare the resample predictions for evaluation
unnest_modeltime_resamples(m750_training_resamples_fitted)
```
Index

* datasets
  m750_training_resamples_fitted, 3
get_target_text_from_resamples, 2
m750_training_resamples_fitted, 3
modelttime::default_forecast_accuracy_metric_set(), 6, 9
modelttime::m750_models, 3
modelttime::m750_training_resamples, 3
modelttime::table_modelttime_accuracy(), 6
modelttime_fit_resamples, 4
modelttime_fit_resamples(), 5, 6, 8
modelttime_resample_accuracy, 5
modelttime_resample_accuracy(), 4, 9
plot_modelttime_resamples, 7
rsample::vfold_cv(), 4
timetk::time_series_cv(), 4
tune::control_resamples(), 4
unnest_modelttime_resamples, 9
unnest_modelttime_resamples(), 2
yardstick::mae(), 6, 9
yardstick::mape(), 6, 9
yardstick::mase(), 6, 9
yardstick::rmse(), 6, 9
yardstick::rsq(), 6, 9
yardstick::smape(), 6, 9