Package 'caretMultimodal'

June 30, 2026

Title Multimodal Late Fusion with 'caret'

Version 1.0.0

Description Extends the 'caret' framework to support late fusion workflows, enabling users to train models independently across multiple data modalities and combine their predictions into a single meta-model. Designed for developers, data scientists, and biomedical researchers alike, 'caretMultimodal' aims to make late fusion ensemble modelling as accessible and flexible as single-dataset workflows in 'caret'. Late fusion methods are based on Wolpert (1992) <doi:10.1016/S0893-6080(05)80023-1>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), randomForest, doParallel

Imports caret, data.table, ggplot2, pROC, foreach, viridis, MultiAssayExperiment, glmnet

Config/testthat/edition 3

URL https://github.com/CompBio-Lab/caretMultimodal, https://compbio-lab.github.io/caretMultimodal/

BugReports https://github.com/CompBio-Lab/caretMultimodal/issues

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Josh Dyce [aut, cre], Amrit Singh [aut]

Maintainer Josh Dyce <jpdyce@gmail.com>

Repository CRAN

Date/Publication 2026-06-30 12:30:07 UTC

Contents

`caret_list` *Construct a* `caret_list` *object*

Description

Builds a list of `caret::train` objects, where each model corresponds to a data set in `data_list`. The resulting list is used as input to `caret_stack()` to construct a meta model.

Usage

```
caret_list(
  target,
  data_list,
  method,
  identifier_column_name = NULL,
  trControl = NULL,
  metric = NULL,
  trim = TRUE,
  do_parallel = TRUE,
  ...
)
```

Arguments

target	Target vector, either numeric for regression or a factor/character for classification.
data_list	A named list of matrix-like objects, where each element is a dataset to train a separate model on. Names are preserved in the returned caret_list and should be unique.
method	The method to train the models with. Can be a custom method or one found in caret::modelLookup().
identifier_column_name	A string giving the name of a column that links rows across datasets (e.g. a participant ID). If provided, this column must be present in all datasets in data_list for proper row matching. Use this when datasets have different numbers of rows. If NULL, datasets are assumed to have identical rows in the same order. Default is NULL. **Note:** Providing identifier_column_name disables support for custom trControl. A new resampling scheme must be constructed for each dataset after row matching, which may invalidate user-supplied fold indices.
trControl	Control for use with the caret::train function. If NULL, .default_control() is used to construct a default control (5-fold cross validation) depending on the target type. Default is NULL.
metric	Metric for use with caret::train function. If NULL, .default_metric() is used to construct a default metric depending on the target type. Default is NULL.
trim	Logical, whether the trained models should be trimmed to save memory. Default is TRUE.
do_parallel	Logical, whether to parallelize model training across datasets. Default is TRUE.
...	Additional arguments to pass to the caret::train function (e.g. tuneGrid = tuneGrid).

Value

A caret_list object (a named list of trained caret::train models corresponding to data_list).

Examples

```
set.seed(42)

data(heart_failure_datasets)

data_list <-  heart_failure_datasets[c("cells", "holter", "mrna", "proteins")]

# Define hyperparameters to tune (optional)
tuneGrid <- expand.grid(alpha = 0.5, lambda = c(0.01, 0.1))

# Construct caret_list object
base_models <- caret_list(
  target = heart_failure_datasets$demo$hospitalizations,
  data_list = data_list,
```

```
  method = "glmnet",
  tuneGrid = tuneGrid
)

class(base_models)
```

---

caret_stack *Construct a* caret_stack *object.*

Description

Train an ensemble (stacked) model from the base learners in a caret_list. The ensemble is itself a caret::train model that learns to combine the predictions of the base models. By default, the meta-learner is trained on out-of-fold predictions from the resampling process, ensuring that the ensemble does not overfit to in-sample predictions. Alternatively, new datasets can be supplied via data_list and target for transfer-learning style ensembling.

Usage

```
caret_stack(
  caret_list,
  method,
  data_list = NULL,
  target = NULL,
  trControl = NULL,
  metric = NULL,
  ...
)
```

Arguments

caret_list	a caret_list object
method	The method to train the ensemble model. Can be a custom method or one found in caret::modelLookup().
data_list	A list of datasets to predict on, with each dataset matching the corresponding model in caret_list. If NULL, the out-of-fold predictions from the base models will be used. Default is NULL.
target	Target parameter vector that must be provided if predicting on a new data list. If NULL, the target vector used to train the base models will be used.
trControl	Control for use with the caret::train function. If NULL, .default_control() is used to construct a default control (5-fold cross validation) depending on the target type. Default is NULL.
metric	Metric for use with caret::train function. If NULL, .default_metric() is used to construct a default metric depending on the target type. Default is NULL.
...	Additional arguments to pass to caret::train

Value

A caret_stack object.

Examples

```
set.seed(42)

data(heart_failure_datasets)

data_list <-  heart_failure_datasets[c("cells", "holter", "mrna", "proteins")]

# Define hyperparameters to tune (optional)
tuneGrid <- expand.grid(alpha = 0.5, lambda = c(0.01, 0.1))

# Construct caret_list object
base_models <- caret_list(
  target = heart_failure_datasets$demo$hospitalizations,
  data_list = data_list,
  method = "glmnet",
  tuneGrid = tuneGrid
)

# Train a Random Forest stacked model on the out-of-fold predictions from the base models
stacked_model <- caret_stack(
  caret_list = base_models,
  method = "rf"
)

class(stacked_model)
```

---

compute_ablation.caret_stack

Perform an ablation analysis for a caret_stack

Description

This function performs an ablation analysis on a caret_stack ensemble to evaluate each base learner's contribution to predictive performance.

Starting from the full ensemble, the procedure iteratively removes one base learner per step. At each step:

1. The ensemble meta-learner is retrained on the remaining base learners, using the same method, tuneGrid, and trControl as the original stack.
2. Variable importance scores are extracted from the retrained meta-learner to estimate each remaining learner's relative contribution.
3. Out-of-fold predictions are generated and scored with metric_function.
4. The learner with the lowest importance score (or highest, if reverse = TRUE) is removed before the next iteration.

Usage

```
## S3 method for class 'caret_stack'
compute_ablation(object, metric_function, metric_name, reverse = FALSE, ...)
```

Arguments

object	A caret_stack object.
metric_function	A function that takes two arguments (predictions, target) and returns a single numeric value representing the metric to compute (e.g., RMSE, accuracy, AUC). predictions are the ensemble's out-of-fold predicted values and target is the response vector.
metric_name	The name of the metric. Used as a row label in the returned data.frame.
reverse	Logical, controls the direction to ablate in. If FALSE, the lowest contributing model is removed at each iteration. If TRUE, the highest contributing model is removed. Default is FALSE.
...	Not used. Included for S3 compatibility.

Value

A data.table

Note

This function does not support for multiclass classifiers.

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

# Since the example stack is a binary classifier,
# this metric function needs to take in predictions (floats) and
# ground truth (binary vector), and produce a single number.
metric_fun <- function(preds, target) {
  pROC::roc(response = target, predictor = preds, quiet = TRUE)$auc
}

compute_ablation(heart_failure_stack, metric_fun, "AUC")
```

---

compute_feature_contributions.caret_stack

Compute the feature level contributions for a caret_stack.

Description

Computes the contribution of each individual feature to the ensemble's predictions using a two-stage application of `caret::varImp`:

1. **Dataset-level weights:** `varImp` is applied to the ensemble meta-learner, treating each base model's predictions as a feature. This yields a relative importance weight for each dataset.
2. **Feature-level importance:** `varImp` is applied to each base model individually, yielding feature importance scores within each dataset.

The final contribution of a feature is the product of its dataset-level weight and its within-dataset feature importance score. All scores are normalized to sum to 100.

Usage

```
## S3 method for class 'caret_stack'
compute_feature_contributions(object, n_features = 20, ...)
```

Arguments

`object`	A `caret_stack` object.
`n_features`	The maximum number of features to include. Setting to a very large value will include all features. Default is 20.
`...`	Not used. Included for S3 compatibility.

Value

A `data.table`

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

compute_feature_contributions(heart_failure_stack)
```

`compute_metric.caret_stack` *Compute metrics with a provided metric function*

Description

The metric_function is applied to the out-of-fold predictions for the caret_stack.

Usage

```
## S3 method for class 'caret_stack'
compute_metric(object, metric_function, metric_name, descending = TRUE, ...)
```

Arguments

object	A caret_stack object
metric_function	A function that takes two arguments (predictions, target) and returns a single numeric value representing the metric to compute (e.g., RMSE, accuracy, AUC).
metric_name	The name of the metric
descending	Whether to sort in descending order. If FALSE, the output is sorted in ascending order. Default is TRUE.
...	Not used. Included for S3 compatibility.

Value

A data.table of metrics

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

# Since the example stack is a binary classifier,
# this metric function needs to take in predictions (floats) and
# ground truth (binary vector), and produce a single number.
metric_fun <- function(preds, target) {
  pROC::roc(response = target, predictor = preds, quiet = TRUE)$auc
}

compute_metric(heart_failure_stack, metric_fun, "AUC")
```

---

compute_model_contributions.caret_stack

Compute the relative contributions of each of the base models in the ensemble model

---

Description

The relative contributions are calculated using the caret::varImp function on the ensemble model. A scaling factor is applied to make the contributions sum to 100%.

Usage

```
## S3 method for class 'caret_stack'
compute_model_contributions(object, descending = TRUE, ...)
```

Arguments

object	A caret_stack object
descending	Whether to sort in descending order. If FALSE, the output is sorted in ascending order. Default is TRUE.
...	Not used. Included for S3 compatibility.

Value

A data.table

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

compute_model_contributions(heart_failure_stack)
```

---

heart_failure_datasets

Heart Failure Datasets

Description

A multimodal dataset from Singh et al. (2019) containing demographic, cellular, electrophysiological, and molecular features for predicting cardiac-related hospitalizations. Used in examples throughout the caretMultimodal package.

Usage

```
heart_failure_datasets
```

Format

A named list with 5 elements:

demo A data.frame of demographic features

cells A data.frame of cell count features

holter A data.frame of Holter monitor (ECG) features

mrna A data.frame of mRNA expression features

proteins A data.frame of protein abundance features

Source

Singh et al. Ensembling Electrical and Proteogenomics Biomarkers for Improved Prediction of Cardiac-Related 3-Month Hospitalizations: A Pilot Study. *Can J Cardiol.* 2019 Apr. doi:10.1016/j.cjca.2018.11.021

heart_failure_stack *Pre-trained* caret_stack *on Heart Failure Datasets*

Description

A caret_stack object pre-trained on heart_failure_datasets. Used in examples throughout the caretMultimodal package.

Usage

```
heart_failure_stack
```

Format

A caret_stack object

See Also

heart_failure_datasets

oof_predictions.caret_list *Out-of-fold predictions from a* caret_list

Description

Retrieve the out-of-fold predictions corresponding to the best hyperparameter setting of the trained caret models. These predictions come from the resampling process (not the final refit) and can optionally be aggregated across resamples to produce a single prediction per training instance.

Usage

```
## S3 method for class 'caret_list'
oof_predictions(
  object,
  drop_redundant_class = TRUE,
  aggregate_resamples = TRUE,
  intersection_only = TRUE,
  ...
)
```

Arguments

object	A caret_list object
drop_redundant_class	Logical, whether to exclude the first class level from prediction output. Default is TRUE.
aggregate_resamples	Logical, whether to aggregate resamples across folds. Default is TRUE.
intersection_only	Logical, whether to trim down the out-of-fold predictions to only the intersection of samples that are present across all models in the list (i.e., the intersection of training indices used during resampling). Default is TRUE.
...	Not used. Included for S3 compatibility.

Value

A data.table::data.table of out-of-fold predictions, with samples as rows and predictions as columns.

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

# Extract the caret_list object from the caret_stack
base_models <- heart_failure_stack$caret_list

oof_predictions(base_models)
```

---

oof_predictions.caret_stack

Out-of-fold predictions from a caret_stack

Description

Retrieve the out-of-fold predictions corresponding to the best hyperparameter setting of a trained ensemble model. These predictions come from the resampling process (not the final refit) and can optionally be aggregated across resamples to produce a single prediction per training instance.

The base model predictions returned here are the training data for the ensemble; depending on model setup, these may be true out-of-fold predictions or simply fitted values. For classification models, the predictions always exclude the first class index.

Usage

```
## S3 method for class 'caret_stack'
oof_predictions(
  object,
  drop_redundant_class = TRUE,
  aggregate_resamples = TRUE,
  ...
)
```

Arguments

object	A caret_stack object
drop_redundant_class	A boolean controlling whether to exclude the first class level from prediction output. Default is TRUE.
aggregate_resamples	Logical, whether to aggregate resamples across folds. Default is TRUE.
...	Not used. Included for S3 compatibility.

Value

A data.table::data.table of OOF predictions

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

oof_predictions(heart_failure_stack)
```

---

plot_ablation.caret_stack

Make a bar plot of an ablation analysis for a caret_stack.

Description

Makes a bar plot from compute_ablation.caret_stack output.

Usage

```
## S3 method for class 'caret_stack'
plot_ablation(object, metric_function, metric_name, reverse = FALSE, ...)
```

Arguments

object	A caret_stack object.
metric_function	A function that takes two arguments (predictions, target) and returns a single numeric value representing the metric to compute (e.g., RMSE, accuracy, AUC). predictions are the ensemble's out-of-fold predicted values and target is the response vector.
metric_name	The name of the metric. Used as a row label in the returned data.frame.
reverse	Logical, controls the direction to ablate in. If FALSE, the lowest contributing model is removed at each iteration. If TRUE, the highest contributing model is removed. Default is FALSE.
...	Not used. Included for S3 compatibility.

Value

A ggplot2 bar plot

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

# Since the example stack is a binary classifier,
# this metric function needs to take in predictions (floats) and
# ground truth (binary vector), and produce a single number.
metric_fun <- function(preds, target) {
  pROC::roc(response = target, predictor = preds, quiet = TRUE)$auc
}

plot_ablation(heart_failure_stack, metric_fun, "AUC")
```

---

plot_feature_contributions.caret_stack

Make a bar plot of feature level for a caret_stack.

Description

Constructs a bar plot with the output of compute_feature_contributions.caret_stack.

Usage

```
## S3 method for class 'caret_stack'
plot_feature_contributions(object, n_features = 20, ...)
```

Arguments

object	A caret_stack object.
n_features	The maximum number of features to include. Setting to a very large value will include all features. Default is 20.
...	Not used. Included for S3 compatibility.

Value

A ggplot2 bar plot.

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

plot_feature_contributions(heart_failure_stack)
```

---

plot_metric.caret_stack

Plot metrics computed with a provided metric function

Description

This function constructs a bar plot with the output of the compute metric method. The bars are ordered by increasing value.

Usage

```
## S3 method for class 'caret_stack'
plot_metric(object, metric_function, metric_name, descending = TRUE, ...)
```

Arguments

object	A caret_stack object
metric_function	A function that takes two arguments (predictions, target) and returns a single numeric value representing the metric to compute (e.g., RMSE, accuracy, AUC).
metric_name	The name of the metric
descending	Whether to sort in descending order. If FALSE, the output is sorted in ascending order. Default is TRUE.
...	Not used. Included for S3 compatibility.

Value

A ggplot2 bar chart

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

# Since the example stack is a binary classifier,
# this metric function needs to take in predictions (floats) and
# ground truth (binary vector), and produce a single number.
metric_fun <- function(preds, target) {
  pROC::roc(response = target, predictor = preds, quiet = TRUE)$auc
}

plot_metric(heart_failure_stack, metric_fun, "AUC")
```

---

plot_model_contributions.caret_stack

Plot the relative contributions of each of the base models in the ensemble model

---

Description

The relative contributions are calculated using the `caret::varImp` function on the ensemble model. A scaling factor is applied to make the contributions sum to 100%.

Usage

```
## S3 method for class 'caret_stack'
plot_model_contributions(object, descending = TRUE, ...)
```

Arguments

`object`	A `caret_stack` object
`descending`	Whether to sort in descending order. If `FALSE`, the output is sorted in ascending order. Default is `TRUE`.
`...`	Not used. Included for S3 compatibility.

Value

A `ggplot2` bar chart

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

plot_model_contributions(heart_failure_stack)
```

plot_roc.caret_stack *Plot ROC curves for individual and ensemble models in a caret_stack*

Description

This function calculates ROC curves for all base models and the ensemble model using the out-of-fold predictions from a caret_stack object. The pROC package is used to compute the ROC curves. ROC curves can only be constructed for binary classifiers.

Usage

```
## S3 method for class 'caret_stack'
plot_roc(object, include_auc = TRUE, ...)
```

Arguments

object	A caret_stack object
include_auc	Whether to include AUC values in the legend. Default is True.
...	Not used. Included for S3 compatibility.

Value

A ggplot2 object

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

plot_roc(heart_failure_stack)
```

predict.caret_list *Predict from a* caret_list

Description

Generate a matrix of predictions from each model in a caret_list. For classification models, probabilities are always returned, with the option to drop one class to avoid multicollinearity in downstream stacking models.

Usage

```
## S3 method for class 'caret_list'
predict(object, data_list, drop_redundant_class = TRUE, ...)
```

Arguments

object	A caret_list object.
data_list	A list of datasets to predict on, with each dataset matching the corresponding model in caret_list.
drop_redundant_class	Logical, whether to exclude the first class level from prediction output. Default is TRUE.
...	Additional arguments to pass to caret::predict

Value

A data.table::data.table of predictions

Examples

```
# Load example data and pre-trained caret_stack object
data(heart_failure_datasets)
data(heart_failure_stack)

# Extract the caret_list object from the caret_stack
base_models <- heart_failure_stack$caret_list

# List of datasets to predict on
data_list <- heart_failure_datasets[c("cells", "holter", "mrna", "proteins")]

predict(base_models, data_list)
```

---

predict.caret_stack *Create a matrix of predictions for a* caret_stack *object.*

Description

Create a matrix of predictions for a caret_stack object.

Usage

```
## S3 method for class 'caret_stack'
predict(object, data_list, drop_redundant_class = TRUE, ...)
```

Arguments

object	A caret_stack object
data_list	A list of datasets to predict on, with each dataset matching the corresponding model in caret_list.
drop_redundant_class	A boolean controlling whether to exclude the first class from prediction output. Default is TRUE.
...	Additional arguments to pass to caret::predict.

Value

A `data.table::data.table` of predictions for base and ensemble models.

Examples

```
# Load example data and pre-trained caret_stack object
data(heart_failure_datasets)
data(heart_failure_stack)

# List of datasets to predict on
data_list <- heart_failure_datasets[c("cells", "holter", "mrna", "proteins")]

predict(heart_failure_stack, data_list)
```

`prepare_mae` *Prepare a MultiAssayExperiment for use with caretMultimodal*

Description

Converts a MultiAssayExperiment object from the MultiAssayExperiment package to a simple list of datasets to pass into `caret_list`.

Usage

```
prepare_mae(mae, transpose = FALSE, ...)
```

Arguments

`mae`	The MultiAssayExperiment object.
`transpose`	Whether to transpose the individual matrices. Samples must correspond to rows for caret_list. Default is FALSE.
`...`	Not used. Included for S3 compatibility.

Value

A named list of matrices.

summary.caret_list *Provide a summary of the best tuning parameters and resampling metrics for all the* caret_list *models.*

Description

Provide a summary of the best tuning parameters and resampling metrics for all the caret_list models.

Usage

```
## S3 method for class 'caret_list'
summary(object, ...)
```

Arguments

object	a caret_list object
...	Not used. Included for S3 compatibility.

Value

A data.table with tunes and metrics from each model.

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

# Extract the caret_list object from the caret_stack
base_models <- heart_failure_stack$caret_list

summary(base_models)
```

summary.caret_stack *Get a summary of a* caret_stack *object*

Description

Get a summary of a caret_stack object

Usage

```
## S3 method for class 'caret_stack'
summary(object, ...)
```

Arguments

object	A caret_stack object
...	Not used. Included for S3 compatibility.

Value

A data.table of methods, tuning parameters and performance metrics for the base and ensemble model

Examples

```
# Load pre-trained example caret_stack object
data(heart_failure_stack)

summary(heart_failure_stack)
```

Index