Package 'iai'

December 6, 2021

Type Package

Title Interface to 'Interpretable AI' Modules

Version 1.7.0

Description An interface to the algorithms of 'Interpretable AI' <https://www.interpretable.ai> from the R programming language. 'Interpretable AI' provides various modules, including 'Optimal Trees' for classification, regression, prescription and survival analysis, 'Optimal Imputation' for missing data imputation and outlier detection, and 'Optimal Feature Selection' for exact sparse regression. The 'iai' package is an open-source project. The 'Interpretable AI' software modules are proprietary products, but free academic and evaluation licenses are available.

URL https://www.interpretable.ai

SystemRequirements Julia (>= 1.0) and Interpretable AI System Image (>= 1.0.0)

License MIT + file LICENSE

Imports JuliaCall (>= 0.17.4), stringr, rlang, lifecycle, rappdirs, ggplot2, cowplot, rjson

RoxygenNote 7.1.1

Suggests testthat, covr, xml2

NeedsCompilation no

Author Jack Dunn [aut, cre],
Ying Zhuo [aut],
Interpretable AI LLC [cph]

Maintainer Jack Dunn <jack@interpretable.ai>

Repository CRAN

Date/Publication 2021-12-06 14:00:02 UTC

R topics documented:

`add_julia_processes` *Add additional Julia worker processes to parallelize workloads*

Description

Julia Equivalent: `Distributed.addprocs!`

Usage

```
add_julia_processes(...)
```

Arguments

`...`	Refer to the Julia documentation for available parameters

Details

For more information, refer to the documentation on parallelization

Examples

```
## Not run: iai::add_julia_processes(3)
```

`all_treatment_combinations` *Return a dataframe containing all treatment combinations of one or more treatment vectors, ready for use as treatment candidates in 'fit_predict¡ or 'predict'*

Description

Julia Equivalent: `IAI.all_treatment_combinations`

Usage

```
all_treatment_combinations(...)
```

Arguments

`...`	A vector of possible options for each treatment

Examples

```
## Not run: iai::all_treatment_combinations(c(1, 2, 3))
```

`apply` *Return the leaf index in a tree model into which each point in the features falls*

Description

Julia Equivalent: `IAI.apply`

Usage

```
apply(lnr, X)
```

Arguments

`lnr`	The learner or grid to query.
`X`	The features of the data.

Examples

```
## Not run: iai::apply(lnr, X)
```

apply_nodes — *Return the indices of the points in the features that fall into each node of a trained tree model*

Description

Julia Equivalent: `IAI.apply_nodes`

Usage

```
apply_nodes(lnr, X)
```

Arguments

`lnr`	The learner or grid to query.
`X`	The features of the data.

Examples

```
## Not run: iai::apply_nodes(lnr, X)
```

as.mixeddata — *Convert a vector of values to IAI mixed data format*

Description

Julia Equivalent: `IAI.make_mixed_data`

Usage

```
as.mixeddata(values, categorical_levels, ordinal_levels = c())
```

Arguments

`values`	The vector of values to convert
`categorical_levels`	The values in `values` to treat as categoric levels
`ordinal_levels`	(optional) The values in `values` to treat as ordinal levels, in the order supplied

Examples

```
## Not run:
df <- iris
set.seed(1)
df$mixed <- rnorm(150)
df$mixed[1:5] <- NA  # Insert some missing values
df$mixed[6:10] <- "Not graded"
df$mixed <- iai::as.mixeddata(df$mixed, c("Not graded"))

## End(Not run)
```

`autoplot.grid_search` *Construct a* R*hrefhttps://ggplot2.tidyverse.org/reference/ggplot.html*`ggplot2::ggplot` *object plotting grid search results for Optimal Feature Selection learners*

Description

Construct a `ggplot2::ggplot` object plotting grid search results for Optimal Feature Selection learners

Usage

```
## S3 method for class 'grid_search'
autoplot(x, type = stop("`type` is required"), ...)
```

Arguments

`x`	The grid search to plot
`type`	The type of plot to construct (either `"validation"` or `"importance"`, for more information refer to the Julia documentation for plotting grid search results)
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: ggplot2::autoplot(grid)
```

`autoplot.roc_curve` *Construct a* R*hrefhttps://ggplot2.tidyverse.org/reference/ggplot.html*`ggplot2::ggplot` *object plotting the ROC curve*

Description

Construct a `ggplot2::ggplot` object plotting the ROC curve

Usage

```
## S3 method for class 'roc_curve'
autoplot(x, ...)
```

Arguments

`x`	The ROC curve to plot
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: ggplot2::autoplot(roc)
```

`autoplot.similarity_comparison` *Construct a* R*hrefhttps://ggplot2.tidyverse.org/reference/ggplot.html*`ggplot2::ggplot` *object plotting the results of the similarity comparison*

Description

Construct a `ggplot2::ggplot` object plotting the results of the similarity comparison

Usage

```
## S3 method for class 'similarity_comparison'
autoplot(x, ...)
```

Arguments

`x`	The similarity comparison to plot
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: ggplot2::autoplot(similarity)
```

`autoplot.stability_analysis` *Construct a* R*hrefhttps://ggplot2.tidyverse.org/reference/ggplot.html*`ggplot2::ggplot` *object plotting the results of the stability analysis*

Description

Construct a `ggplot2::ggplot` object plotting the results of the stability analysis

Usage

```
## S3 method for class 'stability_analysis'
autoplot(x, ...)
```

Arguments

`x`	The stability analysis to plot
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: ggplot2::autoplot(stability)
```

categorical_classification_reward_estimator

Learner for conducting reward estimation with categorical treatments and classification outcomes

Description

Julia Equivalent: IAI.CategoricalClassificationRewardEstimator

Usage

```
categorical_classification_reward_estimator(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::categorical_classification_reward_estimator()
```

categorical_regression_reward_estimator

Learner for conducting reward estimation with categorical treatments and regression outcomes

Description

Julia Equivalent: IAI.CategoricalRegressionRewardEstimator

Usage

```
categorical_regression_reward_estimator(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::categorical_regression_reward_estimator()
```

---

categorical_reward_estimator

Learner for conducting reward estimation with categorical treatments

---

Description

This function was deprecated in iai 1.6.0, and [categorical_classification_reward_estimator()] or [categorical_classification_reward_estimator()] should be used instead.

Usage

```
categorical_reward_estimator(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Details

This deprecation is no longer supported as of the IAI v3 release.

IAI Compatibility

Requires IAI version 2.0, 2.1 or 2.2.

Examples

```
## Not run: lnr <- iai::categorical_reward_estimator()
```

categorical_survival_reward_estimator

Learner for conducting reward estimation with categorical treatments and survival outcomes

Description

Julia Equivalent: IAI.CategoricalSurvivalRewardEstimator

Usage

```
categorical_survival_reward_estimator(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::categorical_survival_reward_estimator()
```

cleanup_installation

Remove all traces of automatic Julia/IAI installation

Description

Removes files created by install_julia and install_system_image

Usage

```
cleanup_installation()
```

Examples

```
## Not run: iai::cleanup_installation()
```

clone *Return an unfitted copy of a learner with the same parameters*

Description

Julia Equivalent: `IAI.clone`

Usage

```
clone(lnr)
```

Arguments

`lnr`	The learner to copy.

Examples

```
## Not run: new_lnr <- iai::clone(lnr)
```

convert_treatments_to_numeric *Convert 'treatments' from symbol/string format into numeric values.*

Description

Julia Equivalent: `IAI.convert_treatments_to_numeric`

Usage

```
convert_treatments_to_numeric(treatments)
```

Arguments

`treatments`	The treatments to convert

Examples

```
## Not run: iai::convert_treatments_to_numeric(c("1", "2", "3"))
```

copy_splits_and_refit_leaves

Copy the tree split structure from one learner into another and refit the models in each leaf of the tree using the supplied data

Description

Julia Equivalent: `copy_splits_and_refit_leaves!`

Usage

```
copy_splits_and_refit_leaves(new_lnr, orig_lnr, ...)
```

Arguments

`new_lnr`	The learner to modify and refit
`orig_lnr`	The learner from which to copy the tree split structure
`...`	Refer to the Julia documentation for available parameters

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: iai::copy_splits_and_refit_leaves(new_lnr, orig_lnr, ...)
```

decision_path

Return a matrix where entry `(i, j)` *is true if the* `i`*th point in the features passes through the* `j`*th node in a trained tree model.*

Description

Julia Equivalent: `IAI.decision_path`

Usage

```
decision_path(lnr, X)
```

Arguments

`lnr`	The learner or grid to query.
`X`	The features of the data.

Examples

```
## Not run: iai::decision_path(lnr, X)
```

---

delete_rich_output_param

Delete a global rich output parameter

Description

Julia Equivalent: `IAI.delete_rich_output_param!`

Usage

```
delete_rich_output_param(key)
```

Arguments

`key`	The parameter to delete.

Examples

```
## Not run: iai::delete_rich_output_param("simple_layout")
```

---

equal_propensity_estimator

Learner that estimates equal propensity for all treatments.

Description

For use with data from randomized experiments where treatments are known to be randomly assigned.

Usage

```
equal_propensity_estimator(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Details

Julia Equivalent: `IAI.EqualPropensityEstimator`

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::equal_propensity_estimator()
```

`fit` *Fits a model to the training data*

Description

Julia Equivalent: `IAI.fit!`

Usage

```
fit(lnr, X, ...)
```

Arguments

`lnr`	The learner or grid to fit.
`X`	The features of the data.
`...`	Other parameters, including zero or more target vectors as required by the problem type. Refer to the Julia documentation for available parameters.

Examples

```
## Not run:
X <- iris[, 1:4]
y <- iris$Species
grid <- iai::grid_search(
    iai::optimal_tree_classifier(max_depth = 1),
)
iai::fit(grid, X, y)

## End(Not run)
```

fit_and_expand — *Fit an imputation learner with training features and create adaptive indicator features to encode the missing pattern*

Description

Julia Equivalent: `IAI.fit_and_expand!`

Usage

```
fit_and_expand(lnr, X, ...)
```

Arguments

`lnr`	The learner to use for imputation.
`X`	The dataframe in which to impute missing values.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: lnr <- iai::fit_and_expand(lnr, X, type = "finite")
```

fit_cv — *Fits a grid search to the training data with cross-validation*

Description

Julia Equivalent: `IAI.fit_cv!`

Usage

```
fit_cv(grid, X, ...)
```

Arguments

`grid`	The grid to fit.
`X`	The features of the data.
`...`	Other parameters, including zero or more target vectors as required by the problem type. Refer to the Julia documentation for available parameters.

Examples

```
## Not run:
X <- iris[, 1:4]
y <- iris$Species
grid <- iai::grid_search(
    iai::optimal_tree_classifier(max_depth = 1),
)
iai::fit_cv(grid, X, y)

## End(Not run)
```

`fit_predict` *Fit a reward estimation model on features, treatments and outcomes and return predicted counterfactual rewards for each observation, as well as the score of the internal estimators.*

Description

For categorical treatments, returns the estimated reward under each treatment observed in the data. For numeric treatments, returns the estimated reward under each treatment candidate.

Usage

```
fit_predict(lnr, X, treatments, ...)
```

Arguments

`lnr`	The learner or grid to use for estimation
`X`	The features of the data.
`treatments`	The treatment applied to each point in the data.
`...`	Additional arguments depending on the treatment and outcome types. Refer to the Julia documentation for more information.

Details

Julia Equivalent: `IAI.fit_predict!`

Examples

```
## Not run: iai::fit_predict(lnr, X, treatments, outcomes)
```

fit_transform — *Fit an imputation model using the given features and impute the missing values in these features*

Description

Similar to calling `fit` followed by `transform`

Usage

```
fit_transform(lnr, X, ...)
```

Arguments

`lnr`	The learner or grid to use for imputation
`X`	The features of the data.
`...`	Refer to the Julia documentation for available parameters.

Details

Julia Equivalent: `IAI.fit_transform!`

Examples

```
## Not run:
X <- iris
X[1, 1] <- NA
grid <- iai::grid_search(
    iai::imputation_learner(),
    method = c("opt_knn", "opt_tree"),
)
iai::fit_transform(grid, X)

## End(Not run)
```

fit_transform_cv — *Train a grid using cross-validation with features and impute all missing values in these features*

Description

Julia Equivalent: `IAI.fit_transform_cv!`

Usage

```
fit_transform_cv(grid, X, ...)
```

Arguments

grid	The grid to use for imputation
X	The features of the data.
...	Refer to the Julia documentation for available parameters.

Examples

```
## Not run:
X <- iris
X[1, 1] <- NA
grid <- iai::grid_search(
    iai::imputation_learner(),
    method = c("opt_knn", "opt_tree"),
)
iai::fit_transform_cv(grid, X)

## End(Not run)
```

---

get_best_params *Return the best parameter combination from a grid*

Description

Julia Equivalent: `IAI.get_best_params`

Usage

```
get_best_params(grid)
```

Arguments

grid	The grid search to query.

Examples

```
## Not run: iai::get_best_params(grid)
```

`get_classification_label` *Return the predicted label at a node of a tree*

Description

Julia Equivalent: `IAI.get_classification_label`

Usage

```
get_classification_label(lnr, node_index, ...)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::get_classification_label(lnr, 1)
```

`get_classification_proba` *Return the predicted probabilities of class membership at a node of a tree*

Description

Julia Equivalent: `IAI.get_classification_proba`

Usage

```
get_classification_proba(lnr, node_index, ...)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::get_classification_proba(lnr, 1)
```

get_cluster_assignments

Return the indices of the trees assigned to each cluster, under the clustering of a given number of trees

Description

Julia Equivalent: get_cluster_assignments

Usage

```
get_cluster_assignments(stability, num_trees)
```

Arguments

stability	The stability analysis to query
num_trees	The number of trees to include in the clustering

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_cluster_assignments(stability, num_trees)
```

get_cluster_details

Return the centroid information for each cluster, under the clustering of a given number of trees

Description

Julia Equivalent: get_cluster_details

Usage

```
get_cluster_details(stability, num_trees)
```

Arguments

stability	The stability analysis to query
num_trees	The number of trees to include in the clustering

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_cluster_details(stability, num_trees)
```

---

get_cluster_distances *Return the distances between the centroids of each pair of clusters, under the clustering of a given number of trees*

Description

Julia Equivalent: get_cluster_distances

Usage

```
get_cluster_distances(stability, num_trees)
```

Arguments

stability	The stability analysis to query
num_trees	The number of trees to include in the clustering

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_cluster_distances(stability, num_trees)
```

---

get_depth *Get the depth of a node of a tree*

Description

Julia Equivalent: IAI.get_depth

Usage

```
get_depth(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_depth(lnr, 1)
```

---

get_estimation_densities

Return the total kernel density surrounding each treatment candidate for the propensity/outcome estimation problems in a fitted learner.

Description

Julia Equivalent: IAI.get_estimation_densities

Usage

```
get_estimation_densities(lnr, ...)
```

Arguments

lnr	The learner from which to extract densities
...	Refer to the Julia documentation for other parameters

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_estimation_densities(lnr, ...)
```

get_features_used *Return the names of the features used by the learner*

Description

Julia Equivalent: IAI.get_features_used

Usage

```
get_features_used(lnr)
```

Arguments

lnr	The learner to query.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_features_used(lnr)
```

get_grid_results *Return a summary of the results from the grid search*

Description

This function was deprecated and renamed to [get_grid_result_summary()] in iai 1.5.0. This is for consistency with the IAI v2.2.0 Julia release.

Usage

```
get_grid_results(grid)
```

Arguments

grid	The grid search to query.

Examples

```
## Not run: iai::get_grid_results(grid)
```

get_grid_result_details

Return a vector of lists detailing the results of the grid search

Description

Julia Equivalent: `IAI.get_grid_result_details`

Usage

```
get_grid_result_details(grid)
```

Arguments

`grid`	The grid search to query.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_grid_result_details(grid)
```

get_grid_result_summary

Return a summary of the results from the grid search

Description

Julia Equivalent: `IAI.get_grid_result_summary`

Usage

```
get_grid_result_summary(grid)
```

Arguments

`grid`	The grid search to query.

Examples

```
## Not run: iai::get_grid_result_summary(grid)
```

get_learner — *Return the fitted learner using the best parameter combination from a grid*

Description

Julia Equivalent: `IAI.get_learner`

Usage

```
get_learner(grid)
```

Arguments

grid	The grid to query.

Examples

```
## Not run: lnr <- iai::get_learner(grid)
```

get_lower_child — *Get the index of the lower child at a split node of a tree*

Description

Julia Equivalent: `IAI.get_lower_child`

Usage

```
get_lower_child(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_lower_child(lnr, 1)
```

get_machine_id *Return the machine ID for the current computer.*

Description

This ID ties the IAI license file to your machine.

Usage

```
get_machine_id()
```

Examples

```
## Not run: iai::get_machine_id()
```

get_num_fits *Return the number of fits along the path in the trained learner*

Description

Julia Equivalent: `IAI.get_num_fits`

Usage

```
get_num_fits(lnr)
```

Arguments

`lnr`	The GLMNet learner to query.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::get_num_fits(lnr)
```

get_num_nodes *Return the number of nodes in a trained learner*

Description

Julia Equivalent: IAI.get_num_nodes

Usage

```
get_num_nodes(lnr)
```

Arguments

lnr	The learner to query.

Examples

```
## Not run: iai::get_num_nodes(lnr)
```

get_num_samples *Get the number of training points contained in a node of a tree*

Description

Julia Equivalent: IAI.get_num_samples

Usage

```
get_num_samples(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_num_samples(lnr, 1)
```

get_params *Return the value of all parameters on a learner*

Description

Julia Equivalent: `IAI.get_params`

Usage

```
get_params(lnr)
```

Arguments

lnr	The learner to query.

Examples

```
## Not run: iai::get_params(lnr)
```

get_parent *Get the index of the parent node at a node of a tree*

Description

Julia Equivalent: `IAI.get_parent`

Usage

```
get_parent(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_parent(lnr, 2)
```

`get_policy_treatment_outcome`

Return the quality of the treatments at a node of a tree

Description

Julia Equivalent: `IAI.get_policy_treatment_outcome`

Usage

```
get_policy_treatment_outcome(lnr, node_index, ...)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::get_policy_treatment_outcome(lnr, 1)
```

`get_policy_treatment_rank`

Return the treatments ordered from most effective to least effective at a node of a tree

---

Description

Julia Equivalent: `IAI.get_policy_treatment_rank`

Usage

```
get_policy_treatment_rank(lnr, node_index, ...)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::get_policy_treatment_rank(lnr, 1)
```

---

get_prediction_constant

Return the constant term in the prediction in the trained learner

---

Description

Julia Equivalent: `IAI.get_prediction_constant`

Usage

```
get_prediction_constant(lnr, ...)
```

Arguments

`lnr`	The learner to query.
`...`	If a GLMNet learner, the index of the fit in the path to query, defaulting to the best fit if not supplied.

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run: iai::get_prediction_constant(lnr)
```

`get_prediction_weights` *Return the weights for numeric and categoric features used for prediction in the trained learner*

Description

Julia Equivalent: `IAI.get_prediction_weights`

Usage

```
get_prediction_weights(lnr, ...)
```

Arguments

`lnr`	The learner to query.
`...`	If a GLMNet learner, the index of the fit in the path to query, defaulting to the best fit if not supplied.

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run: iai::get_prediction_weights(lnr)
```

`get_prescription_treatment_rank` *Return the treatments ordered from most effective to least effective at a node of a tree*

Description

Julia Equivalent: `IAI.get_prescription_treatment_rank`

Usage

```
get_prescription_treatment_rank(lnr, node_index, ...)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::get_prescription_treatment_rank(lnr, 1)
```

`get_regression_constant`

Return the constant term in the regression prediction at a node of a tree

---

Description

Julia Equivalent: `IAI.get_regression_constant` (for classification, regression or prescription tree learners as appropriate)

Usage

```
get_regression_constant(lnr, node_index, ...)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.
`...`	If a prescription problem, the treatment to query.

Examples

```
## Not run:
iai::get_regression_constant(lnr, 1)
iai::get_regression_constant(lnr, 1, "A")

## End(Not run)
```

`get_regression_weights`

Return the weights for each feature in the regression prediction at a node of a tree

---

Description

Julia Equivalent: `IAI.get_regression_weights` (for classification, regression or prescription tree learners as appropriate)

Usage

```
get_regression_weights(lnr, node_index, ...)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.
...	If a prescription problem, the treatment to query.

Examples

```
## Not run:
iai::get_regression_weights(lnr, 1)
iai::get_regression_weights(lnr, 1, "A")

## End(Not run)
```

---

get_rich_output_params

Return the current global rich output parameter settings

Description

Julia Equivalent: IAI.get_rich_output_params

Usage

```
get_rich_output_params()
```

Examples

```
## Not run: iai::get_rich_output_params()
```

---

get_roc_curve_data

Extract the underlying data from an ROC curve (as returned by R*hrefhttps://docs.interpretable.ai/v3.0.0/IAI-R/reference/#iai::roc_curve*roc_curve*)*

Description

The data is returned as a list with two keys: auc giving the area-under-the-curve, and coords containing a vector of lists representing each point on the curve, each with keys fpr (the false positive rate), tpr (the true positive rate) and threshold (the threshold).

Usage

```
get_roc_curve_data(curve)
```

Arguments

curve	The curve to query.

Details

Julia Equivalent: IAI.get_roc_curve_data

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::get_roc_curve_data(curve)
```

---

get_split_categories *Return the categoric/ordinal information used in the split at a node of a tree*

Description

Julia Equivalent: IAI.get_split_categories

Usage

```
get_split_categories(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_split_categories(lnr, 1)
```

get_split_feature *Return the feature used in the split at a node of a tree*

Description

Julia Equivalent: IAI.get_split_feature

Usage

```
get_split_feature(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_split_feature(lnr, 1)
```

get_split_threshold *Return the threshold used in the split at a node of a tree*

Description

Julia Equivalent: IAI.get_split_threshold

Usage

```
get_split_threshold(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_split_threshold(lnr, 1)
```

get_split_weights *Return the weights for numeric and categoric features used in the hyperplane split at a node of a tree*

Description

Julia Equivalent: IAI.get_split_weights

Usage

```
get_split_weights(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_split_weights(lnr, 1)
```

get_stability_results *Return the trained trees in order of increasing objective value, along with their variable importance scores for each feature*

Description

Julia Equivalent: get_stability_results

Usage

```
get_stability_results(stability)
```

Arguments

stability	The stability analysis to query

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_stability_results(stability)
```

get_survival_curve *Return the survival curve at a node of a tree*

Description

Julia Equivalent: IAI.get_survival_curve

Usage

```
get_survival_curve(lnr, node_index, ...)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.
...	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::get_survival_curve(lnr, 1)
```

get_survival_curve_data *Extract the underlying data from a survival curve (as returned by* Rhrefhttps://docs.interpretable.ai/v3.0.0/IAI-R/reference/#iai::predictpredict *or* Rhrefhttps://docs.interpretable.ai/v3.0.0/IAI-R/reference/#iai::get_survival_curveget_survival_curve*)*

Description

The data is returned as a list with two keys: times containing the time for each breakpoint on the curve, and coefs containing the probability for each breakpoint on the curve.

Usage

```
get_survival_curve_data(curve)
```

Arguments

curve	The curve to query.

Details

Julia Equivalent: IAI.get_survival_curve_data

Examples

```
## Not run: iai::get_survival_curve_data(curve)
```

---

get_survival_expected_time *Return the predicted expected survival time at a node of a tree*

Description

Julia Equivalent: IAI.get_survival_expected_time

Usage

```
get_survival_expected_time(lnr, node_index, ...)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.
...	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::get_survival_expected_time(lnr, 1)
```

---

get_survival_hazard *Return the predicted hazard ratio at a node of a tree*

Description

Julia Equivalent: IAI.get_survival_hazard

Usage

```
get_survival_hazard(lnr, node_index, ...)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.
...	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::get_survival_hazard(lnr, 1)
```

---

get_train_errors *Extract the training objective value for each candidate tree in the comparison, where a lower value indicates a better solution*

Description

Julia Equivalent: get_train_errors

Usage

```
get_train_errors(similarity)
```

Arguments

similarity	The similarity comparison

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_train_errors(similarity)
```

get_tree — *Return a copy of the learner that uses a specific tree rather than the tree with the best training objective.*

Description

Julia Equivalent: `get_tree`

Usage

```
get_tree(lnr, index)
```

Arguments

lnr	The original learner
index	The index of the tree to use

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::get_tree(lnr, index)
```

get_upper_child — *Get the index of the upper child at a split node of a tree*

Description

Julia Equivalent: `IAI.get_upper_child`

Usage

```
get_upper_child(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::get_upper_child(lnr, 1)
```

glmnetcv_classifier *Learner for training GLMNet models for classification problems with cross-validation*

Description

Julia Equivalent: IAI.GLMNetCVClassifier

Usage

```
glmnetcv_classifier(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.3 or higher.

Examples

```
## Not run: lnr <- iai::glmnetcv_classifier()
```

glmnetcv_regressor *Learner for training GLMNet models for regression problems with cross-validation*

Description

Julia Equivalent: IAI.GLMNetCVRegressor

Usage

```
glmnetcv_regressor(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::glmnetcv_regressor()
```

---

glmnetcv_survival_learner *Learner for training GLMNet models for survival problems with cross-validation*

Description

Julia Equivalent: IAI.GLMNetCVSurvivalLearner

Usage

```
glmnetcv_survival_learner(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: lnr <- iai::glmnetcv_survival_learner()
```

---

grid_search *Controls grid search over parameter combinations*

Description

Julia Equivalent: IAI.GridSearch

Usage

```
grid_search(lnr, ...)
```

Arguments

lnr	The learner to use when validating.
...	The parameters to validate over.

Examples

```
## Not run:
grid <- iai::grid_search(
    iai::optimal_tree_classifier(
        random_seed = 1,
    ),
    max_depth = 1:5,
)

## End(Not run)
```

`iai_setup` *Initialize Julia and the IAI package.*

Description

This function is called automatically with default parameters the first time any ‘iai‘ function is used in an R session. If custom parameters for Julia setup are required, this function must be called in every R session before calling other ‘iai‘ functions.

Usage

```
iai_setup(...)
```

Arguments

`...`	All parameters are passed through to `JuliaCall::julia_setup`

Examples

```
## Not run: iai::iai_setup()
```

`imputation_learner` *Generic learner for imputing missing values*

Description

Julia Equivalent: `IAI.ImputationLearner`

Usage

```
imputation_learner(method = "opt_knn", ...)
```

Arguments

method	(optional) Specifies the imputation method to use.
...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::imputation_learner(method = "opt_tree")
```

---

impute — *Impute missing values using either a specified method or through validation*

Description

Julia Equivalent: IAI.impute

Usage

```
impute(X, ...)
```

Arguments

X	The dataframe in which to impute missing values.
...	Refer to the Julia documentation for available parameters.

Details

This function was deprecated in iai 1.7.0. This is for consistency with the IAI v3.0.0 Julia release.

Examples

```
## Not run:
X <- iris
X[1, 1] <- NA
iai::impute(X)

## End(Not run)
```

impute_cv *Impute missing values using cross validation*

Description

Julia Equivalent: IAI.impute_cv

Usage

```
impute_cv(X, ...)
```

Arguments

X	The dataframe in which to impute missing values.
...	Refer to the Julia documentation for available parameters.

Details

This function was deprecated in iai 1.7.0. This is for consistency with the IAI v3.0.0 Julia release.

Examples

```
## Not run:
X <- iris
X[1, 1] <- NA
iai::impute_cv(X, list(method = c("opt_knn", "opt_tree")))

## End(Not run)
```

install_julia *Download and install Julia automatically.*

Description

Download and install Julia automatically.

Usage

```
install_julia(version = "latest", prefix = julia_default_install_dir())
```

Arguments

version	The version of Julia to install (e.g. "1.6.3"). Defaults to "latest", which will install the most recent stable release.
prefix	The directory where Julia will be installed. Defaults to a location determined by rappdirs::user_data_dir.

Examples

```
## Not run: iai::install_julia()
```

---

install_system_image *Download and install the IAI system image automatically.*

Description

Download and install the IAI system image automatically.

Usage

```
install_system_image(
  version = "latest",
  replace_default = F,
  prefix = sysimage_default_install_dir()
)
```

Arguments

version	The version of the IAI system image to install (e.g. "2.1.0"). Defaults to "latest", which will install the most recent release.
replace_default	Whether to replace the default Julia system image with the downloaded IAI system image. Defaults to FALSE.
prefix	The directory where the IAI system image will be installed. Defaults to a location determined by rappdirs::user_data_dir.

Examples

```
## Not run: iai::install_system_image()
```

---

is_categoric_split *Check if a node of a tree applies a categoric split*

Description

Julia Equivalent: IAI.is_categoric_split

Usage

```
is_categoric_split(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::is_categoric_split(lnr, 1)
```

---

is_hyperplane_split *Check if a node of a tree applies a hyperplane split*

Description

Julia Equivalent: IAI.is_hyperplane_split

Usage

```
is_hyperplane_split(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::is_hyperplane_split(lnr, 1)
```

---

is_leaf *Check if a node of a tree is a leaf*

Description

Julia Equivalent: IAI.is_leaf

Usage

```
is_leaf(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::is_leaf(lnr, 1)
```

---

is_mixed_ordinal_split

Check if a node of a tree applies a mixed ordinal/categoric split

---

Description

Julia Equivalent: `IAI.is_mixed_ordinal_split`

Usage

```
is_mixed_ordinal_split(lnr, node_index)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.

Examples

```
## Not run: iai::is_mixed_ordinal_split(lnr, 1)
```

---

is_mixed_parallel_split

Check if a node of a tree applies a mixed parallel/categoric split

---

Description

Julia Equivalent: `IAI.is_mixed_parallel_split`

Usage

```
is_mixed_parallel_split(lnr, node_index)
```

Arguments

`lnr`	The learner to query.
`node_index`	The node in the tree to query.

Examples

```
## Not run: iai::is_mixed_parallel_split(lnr, 1)
```

is_ordinal_split *Check if a node of a tree applies a ordinal split*

Description

Julia Equivalent: IAI.is_ordinal_split

Usage

```
is_ordinal_split(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::is_ordinal_split(lnr, 1)
```

is_parallel_split *Check if a node of a tree applies a parallel split*

Description

Julia Equivalent: IAI.is_parallel_split

Usage

```
is_parallel_split(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::is_parallel_split(lnr, 1)
```

mean_imputation_learner

Learner for conducting mean imputation

Description

Julia Equivalent: IAI.MeanImputationLearner

Usage

```
mean_imputation_learner(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::mean_imputation_learner()
```

missing_goes_lower

Check if points with missing values go to the lower child at a split node of of a tree

Description

Julia Equivalent: IAI.missing_goes_lower

Usage

```
missing_goes_lower(lnr, node_index)
```

Arguments

lnr	The learner to query.
node_index	The node in the tree to query.

Examples

```
## Not run: iai::missing_goes_lower(lnr, 1)
```

`multi_questionnaire` *Generic function for constructing an interactive questionnaire using multiple tree learners*

Description

Julia Equivalent: `IAI.MultiQuestionnaire`

Usage

```
multi_questionnaire(obj, ...)
```

Arguments

`obj`	The object controlling which method is used
`...`	Arguments depending on the specific method used

`multi_questionnaire.default` *Construct an interactive questionnaire using multiple tree learners as specified by questions*

Description

Refer to the documentation on advanced tree visualization for more information.

Usage

```
## Default S3 method:
multi_questionnaire(obj, ...)
```

Arguments

`obj`	The questions to visualize. Refer to the Julia documentation on multi-learner visualizations for more information.
`...`	Additional arguments (unused)

Details

Julia Equivalent: `IAI.MultiQuestionnaire`

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run:
iai::multi_questionnaire(list("Questionnaire for" = list(
   "first learner" = lnr1,
   "second learner" = lnr2
)))

## End(Not run)
```

---

multi_questionnaire.grid_search

Construct an interactive tree questionnaire using multiple tree learners from the results of a grid search

---

Description

Julia Equivalent: `IAI.MultiQuestionnaire`

Usage

```
## S3 method for class 'grid_search'
multi_questionnaire(obj, ...)
```

Arguments

`obj`	The grid to visualize
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::multi_questionnaire(grid)
```

`multi_tree_plot` *Generic function for constructing an interactive tree visualization of multiple tree learners*

Description

Julia Equivalent: `IAI.MultiTreePlot`

Usage

```
multi_tree_plot(obj, ...)
```

Arguments

`obj`	The object controlling which method is used
`...`	Arguments depending on the specific method used

`multi_tree_plot.default` *Construct an interactive tree visualization of multiple tree learners as specified by questions*

Description

Refer to the documentation on advanced tree visualization for more information.

Usage

```
## Default S3 method:
multi_tree_plot(obj, ...)
```

Arguments

`obj`	The questions to visualize. Refer to the Julia documentation on multi-learner visualizations for more information.
`...`	Additional arguments (unused)

Details

Julia Equivalent: `IAI.MultiTreePlot`

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run:
iai::multi_tree_plot(list("Visualizing" = list(
   "first learner" = lnr1,
   "second learner" = lnr2
)))

## End(Not run)
```

---

multi_tree_plot.grid_search

Construct an interactive tree visualization of multiple tree learners from the results of a grid search

---

Description

Julia Equivalent: `IAI.MultiTreePlot`

Usage

```
## S3 method for class 'grid_search'
multi_tree_plot(obj, ...)
```

Arguments

`obj`	The grid to visualize
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::multi_tree_plot(grid)
```

`numeric_classification_reward_estimator` *Learner for conducting reward estimation with numeric treatments and classification outcomes*

Description

Julia Equivalent: `IAI.NumericClassificationRewardEstimator`

Usage

```
numeric_classification_reward_estimator(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::numeric_classification_reward_estimator()
```

`numeric_regression_reward_estimator` *Learner for conducting reward estimation with numeric treatments and regression outcomes*

Description

Julia Equivalent: `IAI.NumericRegressionRewardEstimator`

Usage

```
numeric_regression_reward_estimator(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::numeric_regression_reward_estimator()
```

`numeric_reward_estimator`
Learner for conducting reward estimation with numeric treatments

---

Description

This function was deprecated in iai 1.6.0, and [numeric_classification_reward_estimator()] or [numeric_classification_reward_estimator()] should be used instead.

Usage

```
numeric_reward_estimator(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Details

This deprecation is no longer supported as of the IAI v3 release.

IAI Compatibility

Requires IAI version 2.1 or 2.2.

Examples

```
## Not run: lnr <- iai::numeric_reward_estimator()
```

`numeric_survival_reward_estimator`
Learner for conducting reward estimation with numeric treatments and survival outcomes

---

Description

Julia Equivalent: `IAI.NumericSurvivalRewardEstimator`

Usage

```
numeric_survival_reward_estimator(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::numeric_survival_reward_estimator()
```

`optimal_feature_selection_classifier`
Learner for conducting Optimal Feature Selection on classification problems

---

Description

Julia Equivalent: `IAI.OptimalFeatureSelectionClassifier`

Usage

```
optimal_feature_selection_classifier(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run: lnr <- iai::optimal_feature_selection_classifier()
```

`optimal_feature_selection_regressor`
Learner for conducting Optimal Feature Selection on regression problems

---

Description

Julia Equivalent: `IAI.OptimalFeatureSelectionRegressor`

Usage

```
optimal_feature_selection_regressor(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run: lnr <- iai::optimal_feature_selection_regressor()
```

`optimal_tree_classifier` *Learner for training Optimal Classification Trees*

Description

Julia Equivalent: `IAI.OptimalTreeClassifier`

Usage

```
optimal_tree_classifier(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::optimal_tree_classifier()
```

`optimal_tree_policy_maximizer` *Learner for training Optimal Policy Trees where the policy should aim to maximize outcomes*

Description

Julia Equivalent: `IAI.OptimalTreePolicyMaximizer`

Usage

```
optimal_tree_policy_maximizer(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: lnr <- iai::optimal_tree_policy_maximizer()
```

optimal_tree_policy_minimizer

Learner for training Optimal Policy Trees where the policy should aim to minimize outcomes

Description

Julia Equivalent: `IAI.OptimalTreePolicyMinimizer`

Usage

```
optimal_tree_policy_minimizer(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: lnr <- iai::optimal_tree_policy_minimizer()
```

optimal_tree_prescription_maximizer

Learner for training Optimal Prescriptive Trees where the prescriptions should aim to maximize outcomes

Description

Julia Equivalent: `IAI.OptimalTreePrescriptionMaximizer`

Usage

```
optimal_tree_prescription_maximizer(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::optimal_tree_prescription_maximizer()
```

`optimal_tree_prescription_minimizer`
Learner for training Optimal Prescriptive Trees where the prescriptions should aim to minimize outcomes

---

Description

Julia Equivalent: `IAI.OptimalTreePrescriptionMinimizer`

Usage

```
optimal_tree_prescription_minimizer(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::optimal_tree_prescription_minimizer()
```

`optimal_tree_regressor`
Learner for training Optimal Regression Trees

Description

Julia Equivalent: `IAI.OptimalTreeRegressor`

Usage

```
optimal_tree_regressor(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::optimal_tree_regressor()
```

optimal_tree_survival_learner

Learner for training Optimal Survival Trees

Description

Julia Equivalent: `IAI.OptimalTreeSurvivalLearner`

Usage

```
optimal_tree_survival_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::optimal_tree_survival_learner()
```

optimal_tree_survivor *Learner for training Optimal Survival Trees*

Description

This function was deprecated and renamed to `optimal_tree_survival_learner()` in iai 1.3.0. This is for consistency with the IAI v2.0.0 Julia release.

Usage

```
optimal_tree_survivor(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::optimal_tree_survivor()
```

`opt_knn_imputation_learner`

Learner for conducting optimal k-NN imputation

Description

Julia Equivalent: `IAI.OptKNNImputationLearner`

Usage

```
opt_knn_imputation_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::opt_knn_imputation_learner()
```

`opt_svm_imputation_learner`

Learner for conducting optimal SVM imputation

Description

Julia Equivalent: `IAI.OptSVMImputationLearner`

Usage

```
opt_svm_imputation_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::opt_svm_imputation_learner()
```

opt_tree_imputation_learner *Learner for conducting optimal tree-based imputation*

Description

Julia Equivalent: `IAI.OptTreeImputationLearner`

Usage

```
opt_tree_imputation_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::opt_tree_imputation_learner()
```

plot.grid_search *Plot a grid search results for Optimal Feature Selection learners*

Description

Plot a grid search results for Optimal Feature Selection learners

Usage

```
## S3 method for class 'grid_search'
plot(x, ...)
```

Arguments

`x`	The grid search to plot
`...`	Additional arguments (passed to `ggplot2::autoplot.grid_search`)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: plot(grid)
```

plot.roc_curve *Plot an ROC curve*

Description

Plot an ROC curve

Usage

```
## S3 method for class 'roc_curve'
plot(x, ...)
```

Arguments

x	The ROC curve to plot
...	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: plot(roc)
```

plot.similarity_comparison *Plot a similarity comparison*

Description

Plot a similarity comparison

Usage

```
## S3 method for class 'similarity_comparison'
plot(x, ...)
```

Arguments

x	The similarity comparison to plot
...	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: plot(similarity)
```

`plot.stability_analysis` *Plot a stability analysis*

Description

Plot a stability analysis

Usage

```
## S3 method for class 'stability_analysis'
plot(x, ...)
```

Arguments

`x`	The stability analysis to plot
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: plot(stability)
```

predict

Return the predictions made by the model for each point in the features

Description

Julia Equivalent: `IAI.predict`

Usage

```
predict(lnr, X, ...)
```

Arguments

`lnr`	The learner or grid to use for prediction.
`X`	The features of the data.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::predict(lnr, X)
```

predict_expected_survival_time

Return the expected survival time estimate made by a model for each point in the features.

Description

Julia Equivalent: `IAI.predict_expected_survival_time`

Usage

```
predict_expected_survival_time(lnr, X, ...)
```

Arguments

`lnr`	The learner or grid to use for prediction.
`X`	The features of the data.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::predict_expected_survival_time(lnr, X)
```

`predict_hazard` *Return the fitted hazard coefficient estimate made by a model for each point in the features.*

Description

A higher hazard coefficient estimate corresponds to a smaller predicted survival time.

Usage

```
predict_hazard(lnr, X, ...)
```

Arguments

`lnr`	The learner or grid to use for prediction.
`X`	The features of the data.
`...`	Refer to the Julia documentation for available parameters.

Details

Julia Equivalent: `IAI.predict_hazard`

IAI Compatibility

Requires IAI version 1.2 or higher.

Examples

```
## Not run: iai::predict_hazard(lnr, X)
```

predict_outcomes — *Return the predicted outcome for each treatment made by a model for each point in the features*

Description

Julia Equivalent: `IAI.predict_outcomes` (for prescription or policy learners as appropriate)

Usage

```
predict_outcomes(lnr, X, ...)
```

Arguments

`lnr`	The learner or grid to use for prediction.
`X`	The features of the data.
`...`	For policy learners only, the reward matrix.

IAI Compatibility

Requires IAI version 2.0 or higher for policy learners.

Examples

```
## Not run: iai::predict_outcomes(lnr, X, ...)
```

predict_proba — *Return the probabilities of class membership predicted by a model for each point in the features*

Description

Julia Equivalent: `IAI.predict_proba`

Usage

```
predict_proba(lnr, X)
```

Arguments

`lnr`	The learner or grid to use for prediction.
`X`	The features of the data.

Examples

```
## Not run: iai::predict_proba(lnr, X)
```

predict_reward — *Return counterfactual rewards estimated using learner parameters for each observation in the supplied data and predictions*

Description

Julia Equivalent: `IAI.predict_reward`

Usage

```
predict_reward(lnr, ...)
```

Arguments

`lnr`	The learner or grid to use for estimation
`...`	Additional arguments depending on the treatment and outcome types. Refer to the Julia documentation for more information.

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: iai::predict_reward(lnr, treatments, outcomes, predictions)
```

predict_shap — *Calculate SHAP values for all points in the features using the learner*

Description

Julia Equivalent: `IAI.predict_shap`

Usage

```
predict_shap(lnr, X)
```

Arguments

`lnr`	The XGBoost learner or grid to use for prediction.
`X`	The features of the data.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::predict_shap(lnr, X)
```

`predict_treatment_outcome`

Return the estimated quality of each treatment in the trained model of the learner for each point in the features

---

Description

Julia Equivalent: `IAI.predict_treatment_outcome`

Usage

```
predict_treatment_outcome(lnr, X)
```

Arguments

`lnr`	The learner or grid to use for prediction.
`X`	The features of the data.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::predict_treatment_outcome(lnr, X)
```

`predict_treatment_rank`

Return the treatments in ranked order of effectiveness for each point in the features

---

Description

Julia Equivalent: `IAI.predict_treatment_rank`

Usage

```
predict_treatment_rank(lnr, X)
```

Arguments

lnr	The learner or grid to use for prediction.
X	The features of the data.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::predict_treatment_rank(lnr, X)
```

---

print_path *Print the decision path through the learner for each sample in the features*

Description

Julia Equivalent: IAI.print_path

Usage

```
print_path(lnr, X, ...)
```

Arguments

lnr	The learner or grid to query.
X	The features of the data.
...	Refer to the Julia documentation for available parameters.

Examples

```
## Not run:
iai::print_path(lnr, X)
iai::print_path(lnr, X, 1)

## End(Not run)
```

prune_trees — *Use the trained trees in a learner along with the supplied validation data to determine the best value for the 'cp' parameter and then prune the trees according to this value*

Description

Julia Equivalent: `prune_trees!`

Usage

```
prune_trees(lnr, ...)
```

Arguments

`lnr`	The learner to prune
`...`	Refer to the Julia documentation for available parameters

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: iai::prune_trees(lnr, ...)
```

questionnaire — *Specify an interactive questionnaire of a tree learner*

Description

Julia Equivalent: `IAI.Questionnaire`

Usage

```
questionnaire(lnr, ...)
```

Arguments

`lnr`	The learner to visualize.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run: iai::questionnaire(lnr)
```

`random_forest_classifier` *Learner for training random forests for classification problems*

Description

Julia Equivalent: `IAI.RandomForestClassifier`

Usage

```
random_forest_classifier(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::random_forest_classifier()
```

`random_forest_regressor` *Learner for training random forests for regression problems*

Description

Julia Equivalent: `IAI.RandomForestRegressor`

Usage

```
random_forest_regressor(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::random_forest_regressor()
```

---

random_forest_survival_learner

Learner for training random forests for survival problems

Description

Julia Equivalent: `IAI.RandomForestSurvivalLearner`

Usage

```
random_forest_survival_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::random_forest_survival_learner()
```

---

rand_imputation_learner

Learner for conducting random imputation

Description

Julia Equivalent: `IAI.RandImputationLearner`

Usage

```
rand_imputation_learner(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::rand_imputation_learner()
```

---

read_json *Read in a learner or grid saved in JSON format*

Description

Julia Equivalent: `IAI.read_json`

Usage

```
read_json(filename)
```

Arguments

filename	The location of the JSON file.

Examples

```
## Not run: obj <- iai::read_json("out.json")
```

---

refit_leaves *Refit the models in the leaves of a trained learner using the supplied data*

Description

Julia Equivalent: `refit_leaves!`

Usage

```
refit_leaves(lnr, ...)
```

Arguments

lnr	The learner to refit
...	Refer to the Julia documentation for available parameters

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: iai::refit_leaves(lnr, ...)
```

---

reset_display_label *Reset the predicted probability displayed to be that of the predicted label when visualizing a learner*

Description

Julia Equivalent: `IAI.reset_display_label!`

Usage

```
reset_display_label(lnr)
```

Arguments

`lnr`	The learner to modify.

Examples

```
## Not run: iai::reset_display_label(lnr)
```

---

reward_estimator *Learner for conducting reward estimation with categorical treatments*

Description

This function was deprecated and renamed to `categorical_reward_estimator()` in iai 1.4.0. This is for consistency with the IAI v2.1.0 Julia release.

Usage

```
reward_estimator(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Details

This deprecation is no longer supported as of the IAI v3 release.

IAI Compatibility

Requires IAI version 2.2 or lower.

Examples

```
## Not run: lnr <- iai::reward_estimator()
```

---

roc_curve *Generic function for constructing an ROC curve*

Description

Julia Equivalent: `IAI.ROCCurve`

Usage

```
roc_curve(obj, ...)
```

Arguments

`obj`	The object controlling which method is used
`...`	Arguments depending on the specific method used

---

roc_curve.default *Construct an ROC curve from predicted probabilities and true labels*

Description

Julia Equivalent: `IAI.ROCCurve`

Usage

```
## Default S3 method:
roc_curve(obj, y, positive_label = stop("`positive_label` is required"), ...)
```

Arguments

`obj`	The predicted probabilities for each point in the data.
`y`	The true labels of the data.
`positive_label`	The label for which probability is being predicted.
`...`	Additional arguments (unused)

IAI Compatibility

Requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::roc_curve(probs, y, positive_label=positive_label)
```

---

roc_curve.learner *Construct an ROC curve using a trained model on the given data*

Description

Julia Equivalent: `IAI.ROCCurve`

Usage

```
## S3 method for class 'learner'
roc_curve(obj, X, y, ...)
```

Arguments

`obj`	The learner or grid to use for prediction.
`X`	The features of the data.
`y`	The labels of the data.
`...`	Additional arguments (unused)

Examples

```
## Not run: iai::roc_curve(lnr, X, y)
```

---

score *Generic function for calculating scores*

Description

Julia Equivalent: `IAI.score`

Usage

```
score(obj, ...)
```

Arguments

`obj`	The object controlling which method is used
`...`	Arguments depending on the specific method used

score.default — *Calculate the score for a set of predictions on the given data*

Description

Julia Equivalent: `IAI.score`

Usage

```
## Default S3 method:
score(obj, predictions, truths, ...)
```

Arguments

`obj`	The type of problem.
`predictions`	The predictions to evaluate.
`truths`	The true target values for these observations.
`...`	Other parameters, including the criterion. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::score("regression", y_pred, y_true, criterion="mse")
```

score.learner — *Calculate the score for a model on the given data*

Description

Julia Equivalent: `IAI.score`

Usage

```
## S3 method for class 'learner'
score(obj, X, ...)
```

Arguments

`obj`	The learner or grid to evaluate.
`X`	The features of the data.
`...`	Other parameters, including zero or more target vectors as required by the problem type. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::score(obj, X, y)
```

---

set_display_label *Show the probability of a specified label when visualizing a learner*

Description

Julia Equivalent: `IAI.set_display_label!`

Usage

```
set_display_label(lnr, display_label)
```

Arguments

`lnr`	The learner to modify.
`display_label`	The label for which to show probabilities.

Examples

```
## Not run: iai::set_display_label(lnr, "A")
```

---

set_julia_seed *Set the random seed in Julia*

Description

Julia Equivalent: `Random.seed!`

Usage

```
set_julia_seed(seed)
```

Arguments

`seed`	The seed to set

Examples

```
## Not run: iai::set_julia_seed(1)
```

set_params — *Set all supplied parameters on a learner*

Description

Julia Equivalent: `IAI.set_params!`

Usage

```
set_params(lnr, ...)
```

Arguments

`lnr`	The learner to modify.
`...`	The parameters to set on the learner.

Examples

```
## Not run: iai::set_params(lnr, random_seed = 1)
```

set_reward_kernel_bandwidth — *Save a new reward kernel bandwidth inside a learner, and return new reward predictions generated using this bandwidth for the original data used to train the learner.*

Description

Julia Equivalent: `IAI.set_reward_kernel_bandwidth!`

Usage

```
set_reward_kernel_bandwidth(lnr, ...)
```

Arguments

`lnr`	The learner to modify
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::set_reward_kernel_bandwidth(lnr, ...)
```

set_rich_output_param *Sets a global rich output parameter*

Description

Julia Equivalent: `IAI.set_rich_output_param!`

Usage

```
set_rich_output_param(key, value)
```

Arguments

`key`	The parameter to set.
`value`	The value to set

Examples

```
## Not run: iai::set_rich_output_param("simple_layout", TRUE)
```

set_threshold *For a binary classification problem, update the the predicted labels in the leaves of the learner to predict a label only if the predicted probability is at least the specified threshold.*

Description

Julia Equivalent: `IAI.set_threshold!`

Usage

```
set_threshold(lnr, label, threshold, ...)
```

Arguments

`lnr`	The learner to modify.
`label`	The referenced label.
`threshold`	The probability threshold above which `label` will be be predicted.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::set_threshold(lnr, "A", 0.4)
```

show_in_browser — *Show interactive visualization of an object (such as a learner or curve) in the default browser*

Description

Julia Equivalent: `IAI.show_in_browser`

Usage

```
show_in_browser(obj, ...)
```

Arguments

`obj`	The object to visualize.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Showing a grid search requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::show_in_browser(lnr)
```

show_questionnaire — *Show an interactive questionnaire based on a learner in default browser*

Description

Julia Equivalent: `IAI.show_questionnaire`

Usage

```
show_questionnaire(lnr, ...)
```

Arguments

`lnr`	The learner or grid to visualize.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Showing a grid search requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::show_questionnaire(lnr)
```

`similarity_comparison` *Conduct a similarity comparison between the final tree in a learner and all trees in a new learner to consider the tradeoff between training performance and similarity to the original tree*

Description

Refer to the documentation on tree stability for more information.

Usage

```
similarity_comparison(lnr, new_lnr, deviations)
```

Arguments

`lnr`	The original learner
`new_lnr`	The new learner
`deviations`	The deviation between the original tree and each tree in the new learner

Details

Julia Equivalent: `SimilarityComparison`

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::similarity_comparison(lnr, new_lnr, deviations)
```

single_knn_imputation_learner *Learner for conducting heuristic k-NN imputation*

Description

Julia Equivalent: IAI.SingleKNNImputationLearner

Usage

```
single_knn_imputation_learner(...)
```

Arguments

...	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

Examples

```
## Not run: lnr <- iai::single_knn_imputation_learner()
```

split_data *Split the data into training and test datasets*

Description

Julia Equivalent: IAI.split_data

Usage

```
split_data(task, X, ...)
```

Arguments

task	The type of problem.
X	The features of the data.
...	Other parameters, including zero or more target vectors as required by the problem type. Refer to the Julia documentation for available parameters.

Examples

```
## Not run:
X <- iris[, 1:4]
y <- iris$Species
split <- iai::split_data("classification", X, y, train_proportion = 0.75)
train_X <- split$train$X
train_y <- split$train$y
test_X <- split$test$X
test_y <- split$test$y

## End(Not run)
```

---

stability_analysis *Conduct a stability analysis of the trees in a tree learner*

Description

Refer to the documentation on tree stability for more information.

Usage

```
stability_analysis(lnr, ...)
```

Arguments

lnr	The original learner
...	Additional arguments (refer to Julia documentation)

Details

Julia Equivalent: StabilityAnalysis

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::stability_analysis(lnr, ...)
```

transform — *Impute missing values in a dataframe using a fitted imputation model*

Description

Julia Equivalent: `IAI.transform`

Usage

```
transform(lnr, X)
```

Arguments

`lnr`	The learner or grid to use for imputation
`X`	The features of the data.

Examples

```
## Not run: iai::transform(lnr, X)
```

transform_and_expand — *Transform features with a trained imputation learner and create adaptive indicator features to encode the missing pattern*

Description

Julia Equivalent: `IAI.transform_and_expand`

Usage

```
transform_and_expand(lnr, X, ...)
```

Arguments

`lnr`	The learner to use for imputation.
`X`	The dataframe in which to impute missing values.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: lnr <- iai::transform_and_expand(lnr, X, type = "finite")
```

tree_plot *Specify an interactive tree visualization of a tree learner*

Description

Julia Equivalent: IAI.TreePlot

Usage

```
tree_plot(lnr, ...)
```

Arguments

lnr	The learner to visualize.
...	Refer to the Julia documentation on advanced tree visualization for available parameters.

IAI Compatibility

Requires IAI version 1.1 or higher.

Examples

```
## Not run: iai::tree_plot(lnr)
```

tune_reward_kernel_bandwidth *Conduct the reward kernel bandwidth tuning procedure for a range of starting bandwidths and return the final tuned values.*

Description

Julia Equivalent: IAI.tune_reward_kernel_bandwidth

Usage

```
tune_reward_kernel_bandwidth(lnr, ...)
```

Arguments

lnr	The learner to use for tuning the bandwidth
...	Refer to the Julia documentation for other parameters

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::tune_reward_kernel_bandwidth(lnr, ...)
```

`variable_importance` *Generate a ranking of the variables in the learner according to their importance during training. The results are normalized so that they sum to one.*

Description

Julia Equivalent: `IAI.variable_importance`

Usage

```
variable_importance(lnr, ...)
```

Arguments

`lnr`	The learner to query.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::variable_importance(lnr, ...)
```

`variable_importance_similarity` *Calculate similarity between the final tree in a tree learner with all trees in new tree learner using variable importance scores.*

Description

Julia Equivalent: `variable_importance_similarity`

Usage

```
variable_importance_similarity(lnr, new_lnr, ...)
```

Arguments

lnr	The original learner
new_lnr	The new learner
...	Additional arguments (refer to Julia documentation)

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: iai::variable_importance_similarity(lnr, new_lnr)
```

---

write_booster *Write the internal booster saved in the learner to file*

Description

Julia Equivalent: IAI.write_booster

Usage

```
write_booster(filename, lnr)
```

Arguments

filename	Where to save the output.
lnr	The XGBoost learner with the booster to output.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::write_booster(file.path(tempdir(), "out.json"), lnr)
```

write_dot — *Output a learner in* R*hrefhttp://www.graphviz.org/content/dot-language/.dot format*

Description

Julia Equivalent: `IAI.write_dot`

Usage

```
write_dot(filename, lnr, ...)
```

Arguments

`filename`	Where to save the output.
`lnr`	The learner to output.
`...`	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::write_dot(file.path(tempdir(), "tree.dot"), lnr)
```

write_html — *Output a learner as an interactive browser visualization in HTML format*

Description

Julia Equivalent: `IAI.write_html`

Usage

```
write_html(filename, lnr, ...)
```

Arguments

`filename`	Where to save the output.
`lnr`	The learner or grid to output.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Outputting a grid search requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::write_html(file.path(tempdir(), "tree.html"), lnr)
```

---

write_json *Output a learner or grid in JSON format*

Description

Julia Equivalent: IAI.write_json

Usage

```
write_json(filename, obj, ...)
```

Arguments

filename	Where to save the output.
obj	The learner or grid to output.
...	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::write_json(file.path(tempdir(), "out.json"), obj)
```

---

write_pdf *Output a learner as a PDF image*

Description

Julia Equivalent: IAI.write_pdf

Usage

```
write_pdf(filename, lnr, ...)
```

Arguments

filename	Where to save the output.
lnr	The learner to output.
...	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::write_pdf(file.path(tempdir(), "tree.pdf"), lnr)
```

---

write_png *Output a learner as a PNG image*

Description

Julia Equivalent: IAI.write_png

Usage

```
write_png(filename, lnr, ...)
```

Arguments

filename	Where to save the output.
lnr	The learner to output.
...	Refer to the Julia documentation for available parameters.

Examples

```
## Not run: iai::write_png(file.path(tempdir(), "tree.png"), lnr)
```

---

write_questionnaire *Output a learner as an interactive questionnaire in HTML format*

Description

Julia Equivalent: IAI.write_questionnaire

Usage

```
write_questionnaire(filename, lnr, ...)
```

Arguments

filename	Where to save the output.
lnr	The learner or grid to output.
...	Refer to the Julia documentation for available parameters.

IAI Compatibility

Outputting a grid search requires IAI version 2.0 or higher.

Examples

```
## Not run: iai::write_questionnaire(file.path(tempdir(), "questionnaire.html"), lnr)
```

`write_svg` *Output a learner as a SVG image*

Description

Julia Equivalent: `IAI.write_svg`

Usage

```
write_svg(filename, lnr, ...)
```

Arguments

`filename`	Where to save the output.
`lnr`	The learner to output.
`...`	Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: iai::write_svg(file.path(tempdir(), "tree.svg"), lnr)
```

xgboost_classifier *Learner for training XGBoost models for classification problems*

Description

Julia Equivalent: `IAI.XGBoostClassifier`

Usage

```
xgboost_classifier(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::xgboost_classifier()
```

xgboost_regressor *Learner for training XGBoost models for regression problems*

Description

Julia Equivalent: `IAI.XGBoostRegressor`

Usage

```
xgboost_regressor(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.1 or higher.

Examples

```
## Not run: lnr <- iai::xgboost_regressor()
```

`xgboost_survival_learner` *Learner for training XGBoost models for survival problems*

Description

Julia Equivalent: `IAI.XGBoostSurvivalLearner`

Usage

```
xgboost_survival_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 2.2 or higher.

Examples

```
## Not run: lnr <- iai::xgboost_survival_learner()
```

`zero_imputation_learner` *Learner for conducting zero-imputation*

Description

Julia Equivalent: `IAI.ZeroImputationLearner`

Usage

```
zero_imputation_learner(...)
```

Arguments

`...`	Use keyword arguments to set parameters on the resulting learner. Refer to the Julia documentation for available parameters.

IAI Compatibility

Requires IAI version 3.0 or higher.

Examples

```
## Not run: lnr <- iai::zero_imputation_learner()
```

Index