Package 'qshap'

May 9, 2026

Type Package

Title Fast Calculation of Feature Contributions in Boosting Trees

Version 1.0.1

Date 2026-05-03

Description Computes feature-specific R-squared (R2) contributions for boosting tree models using a Shapley-value-based decomposition of the total R-squared in polynomial time. Supports models fitted with 'XGBoost', 'LightGBM', and 'CatBoost', with optimized backend-specific implementations and cached tree summaries suitable for large-scale problems. Multiple visualization tools are included for interpreting and communicating feature contributions. The methodology is described in Jiang, Zhang, and Zhang (2025) <doi:10.48550/arXiv.2407.03515>. Optional 'CatBoost' support uses the R package 'catboost', which is not distributed on CRAN; installation instructions and released binaries are provided by the CatBoost project at <https://catboost.ai/docs/en/concepts/r-installation>.

License GPL (>= 2)

URL https://github.com/catstats/Q-SHAP_R

BugReports https://github.com/catstats/Q-SHAP_R/issues

Imports Rcpp (>= 1.0.14), xgboost (>= 3.1.3.1), parallel, lightgbm, viridisLite, ggplot2, jsonlite, methods, progress

Suggests shiny

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Author Steven He [aut], Zhongli Jiang [aut, cre], Dabao Zhang [aut]

Maintainer Zhongli Jiang <zhongli.jiang.stats@gmail.com>

Repository CRAN

Date/Publication 2026-05-04 07:00:17 UTC

Contents

`as.data.frame.qshap_result` *Coercion method to data.frame for qshap_result*

Description

Coercion method to data.frame for qshap_result

Usage

```
## S3 method for class 'qshap_result'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A qshap_result object
`row.names`	Not used
`optional`	Not used
`...`	Additional arguments (currently unused)

Value

A `data.frame` with columns `feature` (character) and `rsq` (numeric), sorted by `rsq` in decreasing order.

gazer *Create a QSHAP Tree Explainer*

Description

Creates an explainer object for computing feature-specific Shapley values from a trained tree ensemble model. Supports XGBoost, LightGBM, and CatBoost models.

Usage

```
gazer(model, max_depth = NULL, base_score = NULL, ...)
```

Arguments

`model`	A model object of class `xgboost` or `xgb.Booster` from **xgboost**, or class `lgb.Booster` from **lightgbm**
`max_depth`	Maximum depth of trees, extracted from `model` by default.
`base_score`	Base score for predictions, extracted from `model` by default.
`...`	Additional arguments, for future use

Value

A class of `qshap_tree_explainer` object containing the model information and preprocessed tree structures for fast Shapley value computation

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbose = 0)
explainer <- gazer(model)
```

loss *Alias for qshap_loss*

Description

This is a convenience alias for `qshap_loss()` that provides a shorter function name for calculating feature-specific loss contributions.

Usage

```
loss(explainer, x, y, y_mean_ori = NULL)
```

Arguments

explainer	A qshap_tree_explainer object created by gazer()
x	Feature matrix or data frame
y	Response vector
y_mean_ori	Optional pre-computed mean of y (for efficiency)

Value

A matrix of loss contributions with dimensions (n_samples, n_features)

See Also

qshap_loss

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbose = 0)
explainer <- gazer(model)
loss_matrix <- loss(explainer, X, y)
dim(loss_matrix)
```

---

plot.qshap_rsq *Plot method for qshap_rsq objects*

Description

This S3 method enables ‘plot(x, ...)‘ where ‘x‘ is a ‘qshap_rsq‘ object. It dispatches to the visualization functions in ‘vis‘.

Usage

```
## S3 method for class 'qshap_rsq'
plot(
  x,
  y = NULL,
  type = c("rsq", "elbow", "cumu", "gcorr", "hist", "density", "loss"),
  ...
)
```

Arguments

x	A ‘qshap_rsq‘ object.
y	Not used.
type	Plot type: one of "rsq", "elbow", "cumu", "gcorr", "hist", "density", or "loss".
...	Passed to the underlying visualization function.

Value

A ggplot2 object (invisibly).

---

print.qshap_result *Print method for qshap_result*

Description

Print method for qshap_result

Usage

```
## S3 method for class 'qshap_result'
print(x, n = 10, ...)
```

Arguments

x	A qshap_result object
n	Integer number of top features to display (default: 10)
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the qshap_result object to the console.

print.qshap_tree_explainer *Print method for qshap_tree_explainer*

Description

Print method for qshap_tree_explainer

Usage

```
## S3 method for class 'qshap_tree_explainer'
print(x, ...)
```

Arguments

x	A qshap_tree_explainer object
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the qshap_tree_explainer object to the console.

print.simple_tree *Print method for simple_tree*

Description

Print method for simple_tree

Usage

```
## S3 method for class 'simple_tree'
print(x, ...)
```

Arguments

x	A simple_tree object
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the simple_tree object to the console.

print.tree_summary *Print method for tree_summary*

Description

Print method for tree_summary

Usage

```
## S3 method for class 'tree_summary'
print(x, ...)
```

Arguments

x	A tree_summary object
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the tree_summary object to the console.

qshap *Alias for rsq*

Description

This is a convenience alias for rsq() that provides a shorter function name for calculating feature-specific R-squared values.

Usage

```
qshap(
  explainer,
  x,
  y,
  feature_names = NULL,
  local = FALSE,
  nsample = NULL,
  sd_out = TRUE,
  ci_out = TRUE,
  level = 0.95,
  nfrac = NULL,
  random_state = 42,
  ncore = 1L
)
```

Arguments

explainer	A qshap_tree_explainer object created by gazer()
x	Feature matrix or data frame with n samples and p features
y	Response vector of length n
feature_names	Character vector of feature names. If NULL, uses column names from x.
local	Logical; if TRUE, returns both R-squared values and loss matrix
nsample	Optional integer; number of samples to use (random subsample if less than nrow(x))
sd_out	Logical; if TRUE, returns standard deviations of R-squared estimates
ci_out	Logical; if TRUE, returns Wald-style confidence intervals for each feature's R-squared (normal approximation using sd_rsq)
level	Confidence level for the intervals (default 0.95)
nfrac	Optional numeric in (0,1); fraction of samples to use (alternative to nsample)
random_state	Integer seed for reproducible sampling
ncore	Number of cores for parallel processing. Use -1 for all available cores, or a positive integer. Default is 1 (no parallelization)

Value

A qshap_result object; see rsq for details.

See Also

rsq

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbose = 0)
explainer <- gazer(model)
phi_rsq <- qshap(explainer, X, y)
print(phi_rsq)
```

qshap_result *User-friendly constructor for qshap_result*

Description

User-friendly constructor for qshap_result

Usage

```
qshap_result(
  rsq,
  feature_names = NULL,
  total_rsq = NULL,
  n_samples = NULL,
  n_features = NULL,
  loss = NULL
)
```

Arguments

rsq	Numeric vector of feature-specific R-squared values
feature_names	Character vector of feature names (optional)
total_rsq	Numeric total R-squared (sum of feature-specific values)
n_samples	Integer number of samples used
n_features	Integer number of features
loss	Optional loss matrix (n_samples x n_features)

Value

A validated qshap_result object

rsq *Calculate Feature-Specific R-Squared Values*

Description

Computes feature-specific R-squared values using Q-SHAP decomposition, returning a `qshap_result` object with better formatting and additional metadata. The `qshap_result` object includes feature names, total R^2, sample counts, and provides enhanced `print()`, `summary()`, and `as.data.frame()` methods for easier analysis.

Usage

```
rsq(
  explainer,
  x,
  y,
  feature_names = NULL,
  local = FALSE,
  nsample = NULL,
  sd_out = TRUE,
  ci_out = TRUE,
  level = 0.95,
  nfrac = NULL,
  random_state = 42,
  ncore = 1L
)
```

Arguments

explainer	A qshap_tree_explainer object created by gazer()
x	Feature matrix or data frame with n samples and p features
y	Response vector of length n
feature_names	Character vector of feature names. If NULL, uses column names from x.
local	Logical; if TRUE, returns both R-squared values and loss matrix
nsample	Optional integer; number of samples to use (random subsample if less than nrow(x))
sd_out	Logical; if TRUE, returns standard deviations of R-squared estimates
ci_out	Logical; if TRUE, returns Wald-style confidence intervals for each feature's R-squared (normal approximation using sd_rsq)
level	Confidence level for the intervals (default 0.95)
nfrac	Optional numeric in (0,1); fraction of samples to use (alternative to nsample)
random_state	Integer seed for reproducible sampling
ncore	Number of cores for parallel processing. Use -1 for all available cores, or a positive integer. Default is 1 (no parallelization)

Details

This function provides a user-friendly interface for Q-SHAP R^2 computation:

- Automatically extracts feature names from the input data
- Returns a structured object with metadata
- Provides enhanced printing with top features displayed by default
- Includes a comprehensive summary() method
- Can be easily converted to a data frame with as.data.frame()

Value

A qshap_result object containing:

- rsq: Numeric vector of feature-specific R^2 values
- feature_names: Character vector of feature names
- total_rsq: Total R^2 (sum of feature-specific values)
- n_samples: Number of samples
- n_features: Number of features
- loss: Loss matrix (if local=TRUE)

See Also

qshap_result

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbose = 0)
explainer <- gazer(model)
result <- rsq(explainer, X, y)
print(result)
```

---

summary.qshap_result *Summary method for qshap_result*

Description

Summary method for qshap_result

Usage

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

Arguments

object	A qshap_result object
...	Additional arguments (currently unused)

Value

The input object is returned invisibly. Called primarily for its side effect of printing a detailed summary of the qshap_result object to the console.

summary.qshap_rsq *Summary method for qshap_rsq objects*

Description

Provides a summary of the qshap_rsq object, showing the top features by R-squared contribution

Usage

```
## S3 method for class 'qshap_rsq'
summary(object, n = 10, ...)
```

Arguments

object	A qshap_rsq object
n	Integer number of top features to display (default: 10)
...	Additional arguments (currently unused)

Value

The input object is returned invisibly. Called primarily for its side effect of printing a summary of the qshap_rsq object to the console.

summary.qshap_tree_explainer *Summary method for qshap_tree_explainer*

Description

Provides detailed summary information about the explainer

Usage

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

Arguments

object	A qshap_tree_explainer object
...	Additional arguments (currently unused)

Value

The input object is returned invisibly. Called primarily for its side effect of printing a detailed summary of the qshap_tree_explainer object to the console.

Index