Package 'h2otools'

May 8, 2026

Type Package

Title Machine Learning Model Evaluation for 'h2o' Package

Version 0.4

Depends R (>= 3.5.0)

Description Enhances the H2O platform by providing tools for detailed evaluation of machine learning models. It includes functions for bootstrapped performance evaluation, extended F-score calculations, and various other metrics, aimed at improving model assessment.

License MIT + file LICENSE

Encoding UTF-8

Imports h2o (>= 3.34.0.0), curl, boot

RoxygenNote 7.3.2

URL https://github.com/haghish/h2otools, https://www.sv.uio.no/psi/english/people/academic/haghish/

BugReports https://github.com/haghish/h2otools/issues

NeedsCompilation no

Author E. F. Haghish [aut, cre, cph]

Maintainer E. F. Haghish <haghish@hotmail.com>

Repository CRAN

Date/Publication 2025-03-18 10:40:03 UTC

Contents

`automlModelParam` *AutoML Models' Parameters Summary*

Description

Extracts models' parameters from AutoML grid

Usage

```
automlModelParam(model)
```

Arguments

`model`	a h2o AutoML object

Value

a dataframe of models' parameters

Author(s)

E. F. Haghish

Examples

```
## Not run:
if(requireNamespace("h2o")) {
  library(h2o)
  h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
  prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
  prostate <- h2o.importFile(path = prostate_path, header = TRUE)
  y <- "CAPSULE"
  prostate[,y] <- as.factor(prostate[,y])  #convert to factor for classification
  aml <- h2o.automl(y = y,
                    training_frame = prostate,
                    include_algos = "GLM",
                    max_models = 1,
                    max_runtime_secs = 60)

  # extract the model parameters
  model.param <- automlModelParam(aml@leader)
}

## End(Not run)
```

bootImportance *Bootstrap Variable Importance And Averaged Grid Variable Importance*

Description

Evaluates variable importance as well as bootstrapped variable importance for a single model or a grid of models

Usage

```
bootImportance(model, df, metric, n = 100)
```

Arguments

model	a model or a model grid of models trained by h2o machine learning software
df	dataset for testing the model. if "n" is bigger than 1, this dataset will be used for drawing bootstrap samples. otherwise (default), the entire dataset will be used for evaluating the model
metric	character. model evaluation metric to be passed to boot R package. this could be, for example "AUC", "AUCPR", RMSE", etc., depending of the model you have trained. all evaluation metrics provided for your H2O models can be specified here.
n	number of bootstraps

Value

list of mean perforance of the specified metric and other bootstrap results

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
df <- read.csv(prostate_path)

# prepare the dataset for analysis before converting it to h2o frame.
df$CAPSULE <- as.factor(df$CAPSULE)

# convert the dataframe to H2OFrame and run the analysis
prostate.hex <- as.h2o(df)
aml <- h2o.automl(y = "CAPSULE", training_frame = prostate.hex, max_runtime_secs = 30)
```

```
# evaluate the model performance
perf <- h2o.performance(aml@leader, xval = TRUE)

# evaluate bootstrap performance for the training dataset
#    NOTE that the raw data is given not the 'H2OFrame'
perf <- bootPerformance(model = aml@leader, df = df, metric = "RMSE", n = 500)

## End(Not run)
```

---

bootPerformance *bootPerformance*

Description

Evaluate model performance by bootstrapping from training dataset

Usage

```
bootPerformance(model, df, metric, n = 100)
```

Arguments

model	a model trained by h2o machine learning software
df	training, validation, or testing dataset to bootstrap from
metric	character. model evaluation metric to be passed to boot R package. this could be, for example "AUC", "AUCPR", RMSE", etc., depending of the model you have trained. all evaluation metrics provided for your H2O models can be specified here.
n	number of bootstraps

Value

list of mean perforance of the specified metric and other bootstrap results

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
df <- read.csv(prostate_path)

# prepare the dataset for analysis before converting it to h2o frame.
df$CAPSULE <- as.factor(df$CAPSULE)
```

```
# convert the dataframe to H2OFrame and run the analysis
prostate.hex <- as.h2o(df)
aml <- h2o.automl(y = "CAPSULE", training_frame = prostate.hex, max_runtime_secs = 30)

# evaluate the model performance
perf <- h2o.performance(aml@leader, xval = TRUE)

# evaluate bootstrap performance for the training dataset
#    NOTE that the raw data is given not the 'H2OFrame'
perf <- bootPerformance(model = aml@leader, df = df, metric = "RMSE", n = 500)

## End(Not run)
```

`capture` *Capture Evaluation Side Effects*

Description

This function evaluates an R expression while capturing its printed output, messages, warnings, and errors. It returns a list containing the result of the evaluation along with all the captured texts.

Usage

```
capture(expr)
```

Arguments

`expr`	An R expression to evaluate. The expression is captured using `substitute()` to retain its code form before evaluation.

Details

The function uses `withCallingHandlers()` and `tryCatch()` to capture side effects of evaluating the expression. Printed output is captured using `capture.output()`. Warnings and messages are intercepted and their default display is suppressed using `invokeRestart("muffleWarning")` and `invokeRestart("muffleMessage")`, respectively. If an error occurs, its message is stored and `NULL` is returned as the value.

Value

A list with the following components:

value The result of evaluating `expr`.

output A character vector with the printed output produced during evaluation.

messages A character vector with any messages generated during evaluation.

warnings A character vector with any warnings produced during evaluation.

error A character string with the error message if an error occurred; otherwise `NULL`.

Examples

```
## Not run:
  # Example: Capturing output, messages, warnings, and errors
  captured <- capture({
    print("Hello, world!")
    message("This is a message.")
    warning("This is a warning.")
    42  # Final value returned
  })

  # Display the captured components
  print(captured$output)    # Printed output
  print(captured$messages)   # Messages
  print(captured$warnings)   # Warnings
  print(captured$error)      # Error message (if any)
  print(captured$value)      # The evaluated result (42 in this example)

## End(Not run)
```

---

checkFrame *check input data.frame*

Description

checks the class of the input data.frame, makes sure that the specified 'df' is indeed a data.frame and more over, there is no column with class 'character' or 'ordered' in the data.frame. this function helps you ensure that your data is compatible with h2o R package.

Usage

```
checkFrame(df, ignore = NULL, is.df = TRUE, no.char = TRUE, no.ordered = TRUE)
```

Arguments

df	data.frame object to evaluate
ignore	character vector of column names that should be ignored, if any.
is.df	logical. if TRUE, it examines if the 'df' is 'data.frame'
no.char	logical. if TRUE, it examines if the 'df' has any columns of class 'character'
no.ordered	logical. if TRUE, it examines if the 'df' has any columns of class 'ordered' factors

Value

nothing

Author(s)

E. F. Haghish

Examples

```
data(cars)

# no error is expected because 'cars' dataset does not
# have 'ordered' or 'character' columns
checkFrame(cars)
```

`Fmeasure` *F-Measure*

Description

Calculates F-Measure for any given value of Beta

Usage

```
Fmeasure(perf, beta = 1, max = FALSE)
```

Arguments

`perf`	a h2o object of class `"H2OBinomialMetrics"` which is provided by 'h2o.performance' function.
`beta`	numeric, specifying beta value, which must be higher than zero
`max`	logical. default is FALSE. if TRUE, instead of providing the F-Measure for all the thresholds, the highest F-Measure is reported.

Value

a matrix of F-Measures for different thresholds or the highest F-Measure value

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
prostate <- h2o.importFile(path = prostate_path, header = TRUE)
y <- "CAPSULE"
prostate[,y] <- as.factor(prostate[,y])  #convert to factor for classification
aml <- h2o.automl(y = y, training_frame = prostate, max_runtime_secs = 30)
```

```
# evaluate the model performance
perf <- h2o.performance(aml@leader, xval = TRUE)

# evaluate F-Measure for a Beta = 3
Fmeasure(perf, beta = 3, max = TRUE)

# evaluate F-Measure for a Beta = 1.5
Fmeasure(perf, beta = 1.5, max = TRUE)

# evaluate F-Measure for a Beta = 4
Fmeasure(perf, beta = 4, max = TRUE)


## End(Not run)
```

`getPerfMatrix` *getPerfMatrix*

Description

retrieve performance matrix for all thresholds

Usage

```
getPerfMatrix(perf)
```

Arguments

`perf`	a h2o object of class `"H2OBinomialMetrics"` which is provided by 'h2o.performance' function.

Value

a matrix of F-Measures for different thresholds or the highest F-Measure value

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
prostate <- h2o.importFile(path = prostate_path, header = TRUE)
y <- "CAPSULE"
prostate[,y] <- as.factor(prostate[,y])  #convert to factor for classification
aml <- h2o.automl(y = y, training_frame = prostate, max_runtime_secs = 30)
```

```
# evaluate the model performance
perf <- h2o.performance(aml@leader, xval = TRUE)

# get the performance matrix for all thresholds
getPerfMatrix(perf)

## End(Not run)
```

---

h2o.get_ids *h2o.get_ids*

Description

extracts the model IDs from H2O AutoML object or H2O grid

Usage

```
h2o.get_ids(automl)
```

Arguments

automl	a h2o "AutoML" grid object

Value

a character vector of trained models' names (IDs)

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
prostate <- h2o.importFile(path = prostate_path, header = TRUE)
y <- "CAPSULE"
prostate[,y] <- as.factor(prostate[,y])  #convert to factor for classification
aml <- h2o.automl(y = y, training_frame = prostate, max_runtime_secs = 30)

# get the model IDs
ids <- h2o.ids(aml)

## End(Not run)
```

kappa *kappa*

Description

Calculates kappa for all thresholds

Usage

```
kappa(perf, max = FALSE)
```

Arguments

perf	a h2o object of class "H2OBinomialMetrics" which is provided by 'h2o.performance' function.
max	logical. default is FALSE. if TRUE, instead of providing the F-Measure for all the thresholds, the highest F-Measure is reported.

Value

a matrix of F-Measures for different thresholds or the highest F-Measure value

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
prostate <- h2o.importFile(path = prostate_path, header = TRUE)
y <- "CAPSULE"
prostate[,y] <- as.factor(prostate[,y])  #convert to factor for classification
aml <- h2o.automl(y = y, training_frame = prostate, max_runtime_secs = 30)

# evaluate the model performance
perf <- h2o.performance(aml@leader, xval = TRUE)

# evaluate F-Measure for a Beta = 3
kappa(perf, max = TRUE)

## End(Not run)
```

`performance` *provides performance measures using objects from h2o*

Description

takes h2o performance object of class "H2OBinomialMetrics" alongside caret confusion matrix and provides different model performance measures supported by h2o and caret

Usage

```
performance(perf)
```

Arguments

`perf`	h2o performance object of class "H2OBinomialMetrics"

Value

numeric vector

Author(s)

E. F. Haghish

Examples

```
## Not run:
library(h2o)
h2o.init(ignore_config = TRUE, nthreads = 2, bind_to_localhost = FALSE, insecure = TRUE)
prostate_path <- system.file("extdata", "prostate.csv", package = "h2o")
prostate <- h2o.importFile(path = prostate_path, header = TRUE)
y <- "CAPSULE"
prostate[,y] <- as.factor(prostate[,y])  #convert to factor for classification
aml <- h2o.automl(y = y, training_frame = prostate, max_runtime_secs = 30)

# evaluate the model performance
perf <- h2o.performance(aml@leader, xval = TRUE)

# compute more performance measures
performance(perf)


## End(Not run)
```

Index