

Package ‘solitude’

August 24, 2019

Type Package

Title An Implementation of Isolation Forest

Version 0.2.0

Description Isolation forest is anomaly detection method introduced by the paper Isolation based Anomaly Detection (Liu, Ting and Zhou <doi:10.1145/2133360.2133363>).

URL <https://github.com/talegari/solitude>

BugReports <https://github.com/talegari/solitude/issues>

Imports ranger (>= 0.10.0), data.table (>= 1.11.4), igraph (>= 1.2.2),
future.apply (>= 0.2.0), R6 (>= 2.4.0),

Depends R (>= 3.4.0),

Suggests rsample (>= 0.0.4),

License GPL-3

Encoding UTF-8

RoxygenNote 6.1.1

NeedsCompilation no

Author Komala Sheshachala Srikanth [aut, cre]

Maintainer Komala Sheshachala Srikanth <sri.teach@gmail.com>

Repository CRAN

Date/Publication 2019-08-24 20:00:02 UTC

R topics documented:

isolationForest	2
is_integerish	3
solitude	4
terminalNodesDepth	4
terminalNodesDepthPerTree	5
Index	6

`isolationForest`*Fit an Isolation Forest*

Description

'solitude' class implements the isolation forest method introduced by paper Isolation based Anomaly Detection (Liu, Ting and Zhou <doi:10.1145/2133360.2133363>). The extremely randomized trees (extratrees) required to build the isolation forest is grown using [ranger](#) function from **ranger** package.

Usage

```
isolationForest
```

Format

An object of class R6ClassGenerator of length 24.

Design

`$new()` initiates a new 'solitude' object. The possible arguments are:

- `sample_fraction`: (between 0(exclusive) and 1(inclusive), default = 1) Fraction of observations in the dataset to used to build the forest
- `num_trees`: (positive integer, default = 100) Number of trees to be built in the forest
- `replace`: (boolean) Whether the sample of observations should be chosen with replacement when `sample_fraction < 1`
- `seed`: (positive integer) Random seed for the forest
- `nproc`: (NULL or positive integer, default: Use all resources) Number of parallel threads to be used by [ranger](#)
- `respect_unordered_factors`: See `respect.unordered.factors` argument in [ranger](#)

`$fit()` fits a isolation forest for the given dataframe, computes depths of terminal nodes of each tree and stores the anomaly scores and average depth values in `$scores` object as a `data.table`

`$predict()` returns anomaly scores for a new data as a `data.table`

Details

- **Parallelization**: [ranger](#) is parallelized and by default uses all the resources. This is supported when `nproc` is set to NULL. The process of obtaining depths of terminal nodes (which is excuted with `$fit()` is called) may be parallelized separately by setting up a **future** backend.

Examples

```
data("attrition", package = "rsample")
set.seed(1)
index = sample(ceiling(nrow(attrition) * 0.2))
isf = isolationForest$new() # initiate
isf$fit(attrition[index, ]) # fit on 80% data
isf$scores # obtain anomaly scores

# scores closer to 1 might indicate outliers
plot(density(isf$scores$anomaly_score))
round(head(sort(isf$scores$anomaly_score, dec = TRUE), 20), 2)

isf$predict(attrition[-index, ]) # scores for new data
```

is_integerish

Check for a single integer

Description

for a single integer

Usage

```
is_integerish(x)
```

Arguments

x input

Value

TRUE or FALSE

Examples

```
## Not run: is_integerish(1)
```

solitude

An Implementation of Isolation Forest

Description

Isolation forest is an anomaly detection method introduced by the paper Isolation based Anomaly Detection (Liu, Ting and Zhou <doi:10.1145/2133360.2133363>)

Author(s)

Srikanth Komala Sheshachala

See Also

Useful links:

- <https://github.com/talegari/solitude>
 - Report bugs at <https://github.com/talegari/solitude/issues>
-

terminalNodesDepth

Depth of each terminal node of all trees in a ranger model

Description

Depth of each terminal node of all trees in a ranger model is returned as a three column tibble with column names: 'id_tree', 'id_node', 'depth'. Note that root node has the node_id = 0.

Usage

```
terminalNodesDepth(model)
```

Arguments

model A ranger model

Details

This function may be parallelized using a future backend.

Value

A tibble with three columns: 'id_tree', 'id_node', 'depth'.

Examples

```
rf = ranger::ranger(Species ~ ., data = iris, num.trees = 100)
terminalNodesDepth(rf)
```

`terminalNodesDepthPerTree`*Depth of each terminal node of a single tree in a ranger model*

Description

Depth of each terminal node of a single tree in a ranger model. Note that root node has the `node_id = 0`.

Usage

```
terminalNodesDepthPerTree(treelike)
```

Arguments

`treelike` Output of `'ranger::treeInfo'`

Value

Named vector of depths where the names are the terminal node IDs

Examples

```
## Not run:  
rf = ranger::ranger(Species ~ ., data = iris)  
terminalNodesDepthPerTree(ranger::treeInfo(rf, 1))  
  
## End(Not run)
```

Index

*Topic **datasets**

isolationForest, [2](#)

is_integerish, [3](#)

isolationForest, [2](#)

ranger, [2](#)

solitude, [4](#)

solitude-package (solitude), [4](#)

terminalNodesDepth, [4](#)

terminalNodesDepthPerTree, [5](#)