Package ‘dynfeature’

March 1, 2023

Type Package
Title Feature Importance for Dynamic Processes
Version 1.0.1

URL https://github.com/dynverse/dynfeature
BugReports https://github.com/dynverse/dynfeature/issues
License MIT + file LICENSE
Encoding UTF-8
Depends R (>= 3.5.0)
Imports dplyr, dynutils (>= 1.0.2), dynwrap (>= 1.0.0), purrr, methods, ranger, reshape2, testthat, tibble
Suggests caret, covr
RoxygenNote 7.1.1
NeedsCompilation no
Author Robrecht Cannoodt [aut, cre] (<https://orcid.org/0000-0003-3641-729X>, rcannood), Wouter Saelens [aut] (<https://orcid.org/0000-0002-7114-6248>, zouter)
Maintainer Robrecht Cannoodt <rcannood@gmail.com>
Repository CRAN
Date/Publication 2023-03-01 22:30:09 UTC

R topics documented:
calculate_branch_feature_importance .................................................. 2
dynfeature ...................................................................................... 3
fi_ranger_rf_lite ............................................................................. 4

Index 6
calculate_branch_feature_importance

Calculating feature importances across trajectories

Description

Uses the feature importance measures of ranger or caret. calculate_overall_feature_importance calculates the importance for the whole trajectory, calculate_milestone_feature_importance calculates it for individual milestones (e.g. branching points)

Usage

calculate_branch_feature_importance(
  trajectory,
  expression_source = "expression",
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_branching_point_feature_importance(
  trajectory,
  expression_source = "expression",
  milestones_oi = trajectory$milestone_ids,
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_cell_feature_importance(
  trajectory,
  expression_source = "expression",
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_milestone_feature_importance(
  trajectory,
  expression_source = "expression",
  milestones_oi = NULL,
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_overall_feature_importance(
  trajectory,
  expression_source = "expression",
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)
```r
calculate_waypoint_feature_importance(
  trajectory,
  expression_source = "expression",
  waypoints = NULL,
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)
```

**Arguments**

- **trajectory** A trajectory object containing expression values and a trajectory.
- **expression_source** The expression data matrix, with features as columns.
  - If a matrix is provided, it is used as is.
  - If a character is provided, `trajectory[[expression_source]]` should contain the matrix.
  - If a function is provided, that function will be called in order to obtain the expression (useful for lazy loading).
- **verbose** Whether to print out extra information.
- **milestones_oi** The milestone(s) for which to calculate feature importance
- **waypoints** The waypoints, optional

**Value**

A data frame with two or more columns, `feature_id`, and `importance`. `feature_id` is a column in the trajectory expression matrix. Additional columns may be available depending on the function called.

**Examples**

```r
calculate_overall_feature_importance(example_trajectory)
```

---

**dynfeature** *Feature Importance for Dynamic Processes*

**Description**

Calculating feature importance scores from trajectories using the random forests algorithm.
**fi_ranger_rf_lite**  
*Feature Importance methods*

---

**Description**

Feature Importance methods

**Usage**

```r
fi_ranger_rf_lite(
  num_trees = 2000,
  num_variables_per_split = 50,
  num_samples_per_tree = 250,
  min_node_size = 20,
  ...
)
```

```r
fi_ranger_rf(...)
```

```r
fi_caret(caret_method, ...)
```

```r
fi_ranger_rf_tiny(
  num_trees = 100,
  num_variables_per_split = 50,
  num_samples_per_tree = 250,
  min_node_size = 20,
  ...
)
```

---

**Arguments**

- `num_trees` *(fi_ranger_rf_lite)* The number of trees to use
- `num_variables_per_split` *(fi_ranger_rf_lite)* The number of variables to sample per split
- `num_samples_per_tree` *(fi_ranger_rf_lite)* The number of samples to bootstrap per split
- `min_node_size` *(fi_ranger_rf_lite)* The minimum node size, no split will be made if the node size is less than this value.
- `...` Extra parameters to pass onto the underlying feature importance function.
- `caret_method` *(fi_caret)* Which caret method to use for feature importance.

---

**Value**

A list containing a helper function for calling a feature importance function.
Examples

library(dynwrap)
data(example_trajectory)

calculate_overall_feature_importance(example_trajectory, fi_method = fi_ranger_rf())
Index

calculate_branch_feature_importance, 2

calculate_branching_point_feature_importance
  (calculate_branch_feature_importance), 2

calculate_cell_feature_importance
  (calculate_branch_feature_importance), 2

calculate_milestone_feature_importance
  (calculate_branch_feature_importance), 2

calculate_overall_feature_importance
  (calculate_branch_feature_importance), 2

calculate_waypoint_feature_importance
  (calculate_branch_feature_importance), 2

dynfeature, 3

fi_caret (fi_ranger_rf_lite), 4

fi_ranger_rf (fi_ranger_rf_lite), 4

fi_ranger_rf_lite, 4

fi_ranger_rf_tiny (fi_ranger_rf_lite), 4

ranger, 2