Package ‘processpredictR’

January 17, 2023

Type Package
Title Process Prediction
Version 0.1.0
Date 2022-12-23
Description Means to predict process flow, such as process outcome, next activity, next time, remaining time, and remaining trace. Off-the-shelf predictive models based on the concept of Transformers are provided, as well as multiple ways to customize the models. This package is partly based on work described in Zaharah A. Bukhsh, Aaqib Saeed, & Remco M. Dijkman. (2021). “ProcessTransformer: Predictive Business Process Monitoring with Transformer Network” <arXiv:2104.00721>.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.2.3
Imports bupaR, edeaR, dplyr, forcats, magrittr, reticulate, tidyr, tidyselect, purrr, stringr, keras, tensorflow, rlang, data.table, mltools, ggplot2, cli, glue, plotly, progress
Config/testthat/edition 3
Depends R (>= 2.10)
Suggests knitr, rmarkdown, lubridate, eventdataR
VignetteBuilder knitr
NeedsCompilation no
Author Ivan Esin [aut],
Gert Janssenswillen [cre],
Hasselt University [cph]
Maintainer Gert Janssenswillen <gert.janssenswillen@uhasselt.be>
Repository CRAN
Date/Publication 2023-01-17 17:10:01 UTC
R topics documented:

confusion_matrix .................................................. 2
create_model .......................................................... 3
create_vocabulary ................................................... 4
get_vocabulary ....................................................... 4
max_case_length ....................................................... 5
num_outputs .......................................................... 5
plot.ppred_predictions .............................................. 6
ppred_examples_df ................................................... 6
ppred_model ............................................................ 6
ppred_predictions ..................................................... 7
prepare_examples ..................................................... 7
print.ppred_model .................................................... 8
processpredictR ......................................................... 8
split_train_test ....................................................... 9
stack_layers .......................................................... 9
tokenize .............................................................. 10
vocab_size ............................................................ 10

Index 12

confusion_matrix  Confusion matrix for predictions

Description

Confusion matrix for predictions

Usage

confusion_matrix(predictions, ...)

Arguments

predictions ppred_predictions: A data.frame with predicted values returned by predict.ppred_model().
... additional arguments.

Value

A table object that can be used for plotting a confusion matrix using plot().
create_model

**Define transformer model**

**Description**

Defines the model using the keras functional API. The following 4 process monitoring tasks are defined:

- outcome
- next_activity
- next_time
- remaining_time
- remaining_trace
- remaining_trace_s2s

**Usage**

```r
create_model(
  x_train,
  custom = FALSE,
  num_heads = 4,
  output_dim_emb = 36,
  dim_ff = 64,
  ...
)
```

**Arguments**

- **x_train**: data.frame: A processed data.frame from `prepare_examples()`.
- **custom**: logical (default FALSE): If TRUE, returns a custom model.
- **num_heads**: A number of attention heads of the `keras::layer_embedding()`.
- **output_dim_emb**: Dimension of the dense embedding of the `keras::layer_embedding()`.
- **dim_ff**: Dimensionality of the output space of the feedforward network part of the model (units argument of the `keras::layer_dense()`).
- **...**: you can pass additional arguments to `keras::keras_model()` (ex.: name argument).

**Value**

An object of class `ppred_model` and `list` containing a Transformer model (returned by `keras::keras_model()`) and some additional useful metrics.
create_vocabulary  Create a vocabulary

Description

Creates a vocabulary of activities and outcome labels.

Usage

create_vocabulary(processed_df)

Arguments

processed_df  A preprocessed object of type `ppred_examples_df` returned by `prepare_examples()`.

Value

A list consisting of:

- "keys_x": list of activity labels
- "keys_y": list of outcome labels (none for tasks "next_time" and "remaining_time")

get_vocabulary  Utils

Description

Utils

Usage

get_vocabulary(examples)

Arguments

examples  a preprocessed dataset returned by `prepare_examples_dt()`.
max_case_length

Calculate the maximum length of a case / number of activities in the longest trace in an event log.

Usage

max_case_length(processed_df)

Arguments

processed_df A processed dataset of class `ppred_examples_df` returned by `prepare_examples()`.

Value

An integer number of the maximum case length (longest trace) in an event log.

Examples

```r
library(processpredictR)
library(eventdataR)

df <- prepare_examples(patients)
max_case_length(df)
```

num_outputs

Calculate number of outputs (target variables).

Description

Calculate number of outputs (target variables).

Usage

num_outputs(processed_df)

Arguments

processed_df A processed dataset of class `ppred_examples_df`.

Value

An integer number of outputs for supplying as an argument to a Transformer model, i.e. number of unique labels for a specific process monitoring task.
Examples

```r
library(processpredictR)
library(eventdataR)
df <- prepare_examples(patients)
num_outputs(df)
```

---

### `plot.ppred_predictions`

#### Plot Methods

**Description**

Visualize metric

**Usage**

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

**Arguments**

- `x`: Data to plot. An object of type `ppred_predictions`.
- `...`: Additional variables

**Value**

A ggplot object, which can be customized further, if deemed necessary.

---

### `ppred_examples_df`

**Description**

Object of type `ppred_examples_df` is a transformed event log returned by `prepare_examples_dt()`.

---

### `ppred_model`

**Description**

Object of type `ppred_model` is a list returned by `processpredictR::create_model()` containing a custom keras functional (transformer) model and some other useful metrics of an event log.
**ppred_predictions**

**ppred_predictions object**

**Description**

Object of type `ppred_predictions` is a data.frame with predicted values returned by `predict.ppred_model()`.

**prepare_examples**

Convert a dataset of type `log` into a preprocessed format.

**Description**

An event log is converted into a tibble where each row contains a cumulative sequence of activities per case. This sequence will eventually be fed to the Transformer model’s token embedding layer.

**Usage**

```r
prepare_examples(
  log,
  task = c("outcome", "next_activity", "next_time", "remaining_time", "remaining_trace", "remaining_trace_s2s"),
  features = NULL,
  ...
)
```

**Arguments**

- **log**: Object of class `log` or derivatives (`grouped_log`, `eventlog`, `activitylog`, etc.).
- **task**: character: a process monitoring task for which to prepare an event log.
- **features**: character (default NULL): additional features. Appends attributes (if present) numeric_features and/or categorical_features to a preprocessed event log.
- **...**: additional arguments.

**Value**

A preprocessed dataset of class `ppred_examples_df`.

**Examples**

```r
library(processpredictR)
library(eventdataR)

prepare_examples(patients, "next_activity")
```
**Description**

Print methods

**Usage**

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

**Arguments**

- `x` : An object of class `ppred_model`
- `...` : Additional Arguments.

**Value**

prints a Transformer model from a list returned by `create_model()`.

---

**Description**

Means to predict process flow, such as process outcome, next activity, next time, remaining time, and remaining trace. Off-the-shelf predictive models based on the concept of Transformers are provided, as well as multiple ways to customize the models. This package is partly based on work described in Zaharah A. Bukhsh, Aaqib Saeed, & Remco M. Dijkman. (2021). "ProcessTransformer: Predictive Business Process Monitoring with Transformer Network" arXiv:2104.00721.

**Author(s)**

- **Maintainer**: Gert Janssenswillen <gert.janssenswillen@uhasselt.be>
- Authors:
  - Ivan Esin <ivan.esin@student.uhasselt>
- Other contributors:
  - Hasselt University [copyright holder]
split_train_test

Splits the preprocessed data.frame.

Description

Returns train- and test dataframes as a list.

Usage

split_train_test(processed_df, split = 0.7)

Arguments

processed_df A preprocessed object of type ppred_examples_df returned by prepare_examples().
split numeric (default 0.7): A train-test split ratio.

Value

A list containing the train- and the test set objects.

Examples

library(processpredictR)
library(eventdataR)

df <- prepare_examples(patients, "next_activity")
split_train_test(df, split = 0.8)

stack_layers

Stacks a keras layer on top of existing model

Description

User friendly interface to add a keras layer on top of existing model.

Usage

stack_layers(object, ...)

Arguments

object a list containing a model returned by create_model().
... functions for adding layers by using functional keras API. For example, keras::layer_dense(units=32, activation="relu").
Value

a list containing an adapted Transformer model.

tokenize

Tokenize features and target of a processed dataset of class ppred_examples_df

Description

Tokenize features and target of a processed ppred_examples_df object to fit the Transformer model.

Usage

tokenize(processed_df)

Arguments

processed_df A preprocessed object of type ppred_examples_df returned by prepare_examples().

Value

A list of (sequence) tokens and additional numeric or categorical features.

vocab_size

Calculate the vocabulary size, i.e. the sum of number of activities, outcome labels and padding keys

Description

Calculate the vocabulary size, i.e. the sum of number of activities, outcome labels and padding keys

Usage

vocab_size(processed_df)

Arguments

processed_df A processed dataset of class ppred_examples_df from prepare_examples().

Value

an integer number of vocabulary size to define the Transformer model.
Examples

```r
library(processpredictR)
library(eventdataR)
df <- prepare_examples(patients)
vocab_size(df)
```
Index

* visualization
  plot.ppred_predictions, 6
  split_train_test, 9
  stack_layers, 9

activitylog, 7

character, 7

collection_matrix, 2

collection_model, 3

collection_vocabulary, 4

data.frame, 2, 3, 9

eventlog, 7

FALSE, 3

ged_vocabulary, 4

grouped_log, 7

keras::layer_dense(), 3

eras::layer_embedding(), 3

list, 3, 4, 9, 10

log, 7

logical, 3

max_case_length, 5

NULL, 7

num_outputs, 5

numeric, 9

plot.ppred_predictions, 6

ppred_examples_df, 4–6, 6, 7, 9, 10

ppred_model, 3, 6, 6, 8

ppred_predictions, 2, 6, 7, 7

prepare_examples, 7

prepare_examples(), 3

print.ppred_model, 8

processpredictR, 8

processpredictR-package
  (processpredictR), 8

table, 2

tokenize, 10

TRUE, 3

vocab_size, 10

12