enade-py Documentation

Release 0.1.0.post0.dev7+gff405c4

M. Choji

Oct 13, 2020

Contents

This is the documentation of **enade-py**.

CHAPTER 1

Description

enade-py comprises a set of functions for helping researchers and Educational Data Mining (EDM) enthusiasts through the data mining process using Enade microdata.

The Enade microdata datasets are provided by Inep and consist of informations from brazilian undergraduate students and their performance on Enade (a national exam taken at the end of the course).

CHAPTER 2

Contents

2.1 License

2.2 Contributors

- M. Choji <mchoji@users.noreply.github.com>

2.3 Changelog

2.3.1 Current Version

Version 0.1.0, 2020-10-12

- First release

2.4 enadepy

2.4.1 enadepy package

Submodules

enadepy.frequent module

A module for frequent itemsets mining.

`enadepy.frequent.`**`association_rules_ext`**(*freq_itemsets: PandasDataFrame*, ***kwargs*) → PandasDataFrame

Generates association rules from frequent itemsets.

This function extends the function *mlxtend.frequent_patterns.association_rules* by appending information about the length of both antecedent and consequent. If the frequent itemsets have indications of closed frequent itemsets, the output will also set this information for the components of the rule.

Parameters **`freq_itemsets`** (*`PandasDataFrame`*) – A pandas DataFrame containing frequent itemsets.

Returns A pandas DataFrame of association rules including the metrics 'support', 'confidence', 'leverage', 'lift' and 'conviction'.

Return type PandasDataFrame

See also:

freq_itemsets: generates frequent itemsets

mlxtend.frequent_patterns.association_rules

`enadepy.frequent.`**`closed_freq_itemsets`**(*dataframe: PandasDataFrame*, ***kwargs*) → PandasDataFrame

Generates frequent itemsets using FP-Growth.

Generates frequent itemsets as of those generated by *mlxtend.frequent_patterns.fpgrowth* but with two additional columns indicating if the itemset is a closed frequent itemset and its length.

Parameters

- **`dataframe`** (*`PandasDataFrame`*) – A pandas DataFrame in transaction mode.
- **`**kwargs`** (*`Any`*) – Any arguments to be passed to function *mlxtend.frequent_patterns.fpgrowth*.

Returns A pandas DataFrame containing the frequent itemsets with the corresponding lengths and a indication if an itemset is a closed frequent itemset.

Return type PandasDataFrame

See also:

mlxtend.frequent_patterns.fpgrowth

`enadepy.frequent.`**`closed_freq_itemsets_sort`**(*dataframe: PandasDataFrame*, *sort_by: str = 'support'*, *ascending: bool = False*, ***kwargs*) → PandasDataFrame

Generates sorted frequent itemsets using FP-Growth.

Same as closed_freq_itemsets but with output sorted.

Parameters

- **`dataframe`** (*`PandasDataFrame`*) – A pandas DataFrame in transaction mode.
- **`sort_by`** (*`str, optional`*) – The column to use for sorting ('support' or 'length'). Defaults to 'support'.
- **`ascending`** (*`bool, optional`*) – Sort output in ascending mode. Defaults to False.
- **`**kwargs`** (*`Any`*) – Any arguments to be passed to function *mlxtend.frequent_patterns.fpgrowth*.

Returns A pandas DataFrame containing the frequent itemsets with the corresponding lengths and a indication if an itemset is a closed frequent itemset.

Return type PandasDataFrame

See also:

closed_freq_itemsets

`enadepy.frequent.`**`filter_rules`**(*rules: PandasDataFrame, by: List[str] = ['conviction', 'support', 'lift']*) → PandasDataFrame

Excludes duplicated rules according to a given criteria.

This function will sort the rules according to the columns specified and drop rows that contain the same items, considering the union of antecedent and consequent, as of the one with greatest values.

Parameters

- **`rules`** (*`PandasDataFrame`*) – a pandas DataFrame containing association rules.
- **`by`** (*`List[str], optional`*) – A list containing the precedence of columns to be used during rules sorting. Defaults to ['conviction', 'support', 'lift'].

Returns a pandas DataFrame containing filtered rules.

Return type PandasDataFrame

See also:

association_rules_ext, find_itemsets_any, find_itemsets_all

`enadepy.frequent.`**`find_itemsets_all`**(*freq_itemsets: PandasDataFrame*, *search: Set[T] = {}*, *exact: bool = False*, *col_name: str = 'itemsets'*) → PandasDataFrame

Finds itemsets containing all the items given in query.

Parameters

- **`freq_itemsets`** (*`PandasDataFrame`*) – The frequent itemsets where the search will be performed.
- **`search`** (*`Set, optional`*) – Set with items to search for. Defaults to set().
- **`exact`** (*`bool, optional`*) – Match only if itemset is equal to *search*. Defaults to False.

- **col_name** (*str, optional*) – Column name where the itemsets reside. Defaults to 'itemsets'.

Returns a pandas DataFrame containing the itemsets the match requisites.

Return type PandasDataFrame

See also:

find_itemsets_any, find_itemsets_without

enadepy.frequent.**find_itemsets_any**(*freq_itemsets: PandasDataFrame*, *search: Set[T] = {}*, *col_name: str = 'itemsets'*) → PandasDataFrame

Finds itemsets containing any of the items given in query.

Parameters

- **freq_itemsets** (*PandasDataFrame*) – The frequent itemsets where the search will be performed.
- **search** (*Set, optional*) – Set with items to search for. Defaults to set().
- **col_name** (*str, optional*) – Column name where the itemsets reside. Defaults to 'itemsets'.

Returns a pandas DataFrame containing the itemsets the match requisites.

Return type PandasDataFrame

See also:

find_itemsets_all, find_itemsets_without

enadepy.frequent.**find_itemsets_without**(*freq_itemsets: PandasDataFrame*, *search: Set[T] = {}*, *col_name: str = 'itemsets'*) → PandasDataFrame

Finds itemsets that do not contain the items given in query.

Parameters

- **freq_itemsets** (*PandasDataFrame*) – The frequent itemsets where the search will be performed.
- **search** (*Set, optional*) – Set with items to exclude. Defaults to set().
- **col_name** (*str, optional*) – Column name where the itemsets reside. Defaults to 'itemsets'.

Returns a pandas DataFrame containing the itemsets the match requisites.

Return type PandasDataFrame

See also:

find_itemsets_any, find_itemsets_all

enadepy.frequent.**freq_itemsets**(*dataframe: PandasDataFrame*, ***kwargs*) → PandasDataFrame

Generates frequent itemsets from dataframe in transactions mode.

Note: A dataframe in transaction mode is one in which all the columns contain binary values, like True or False.

Parameters

- **dataframe** (*PandasDataFrame*) – A pandas DataFrame in transaction mode.

- **`**kwargs`** (*`Any`*) – Any arguments to be passed to function *mlxtend.frequent_patterns.fpgrowth*.

Returns A pandas DataFrame containing the frequent itemsets with the support and length for each itemset.

Return type PandasDataFrame

`enadepy.frequent.`**`freq_itemsets_sort`**(*dataframe: PandasDataFrame*, *sort_by: str = 'support'*, *ascending: bool = False*, ***kwargs*) → PandasDataFrame

Generates sorted frequent itemsets.

Same as freq_itemsets but with output sorted.

Parameters

- **`dataframe`** (*`PandasDataFrame`*) – A pandas DataFrame in transaction mode.
- **`sort_by`** (*`str, optional`*) – The column to use for sorting (‘support’ or ‘length’). Defaults to ‘support’.
- **`ascending`** (*`bool, optional`*) – Sort output in ascending mode. Defaults to False.
- **`**kwargs`** (*`Any`*) – Any arguments to be passed to function *mlxtend.frequent_patterns.fpgrowth*.

Returns A pandas DataFrame containing the frequent itemsets with the support and length for each itemset.

Return type PandasDataFrame

See also:

freq_itemsets

enadepy.helpers module

A set of helpers for all Enade microdata data mining stages.

`enadepy.helpers.`**`list_cols_disc_status`**(*exclude: List[str] = None*) → List[str]

Returns situation types from discursive questions.

Returns variable names related to the situation types from questions in the discursive part of the exam.

Parameters **`exclude`** (*`List[str], optional`*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_exam`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to the exam.

Parameters **`exclude`** (*`List[str], optional`*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to the exam, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_exam_eval`**(*exclude: List[str] = None*) → List[str]

Returns columns related to the perception about the exame.

Returns variable names related to the perception of the student about the exam.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_grades`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to the grades.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to the grades, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_inst_eval`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to institution evaluation.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to institution evaluation, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_institution`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to the institution.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to the institution, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_licentiate`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to licentiate courses.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to licentiate courses, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_obj_info`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to the objective part of the exam.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to the objective part of the exam, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_presence`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to types of presence.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to types of presence, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_socioecon`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to socioeconomics aspects.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to socioeconomics aspects, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_student`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to the student.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to the student, excluding the ones passed as argument.

Return type List[str]

`enadepy.helpers.`**`list_cols_vectors`**(*exclude: List[str] = None*) → List[str]

Returns variable names related to vectors.

Vectors, in this context, refer to the structures which contain the answers for the questions from the exam.

Parameters **exclude** (*List[str], optional*) – list of variables to exclude from the output. Defaults to None.

Returns The variable names related to vectors, excluding the ones passed as argument.

Return type List[str]

enadepy.index module

A set of indexes that map identifiers to descriptions.

Each index in this module relates to a question or student/institution information (variable) in Enade microdata. Indexes are represented by dictionaries and should not be accessed directly.

`enadepy.index.`**`get_index_dict`**(*varname: str*) → Dict[KT, VT]

Gets a map to translate indexes from a given variable.

Given a variable name (column name from Enade microdata), returns a dictionary containing the values seen in microdata as dictionary's keys and the respective descriptions as dictionary's values.

Parameters **varname** (*str*) – A variable or column name from Enade microdata.

Raises `NameError` – if a dictionary was not found for the given name.

Returns A dictionary mapping values to descriptions for a given variable or column name.

Return type Dict

enadepy.loaders module

Provides functions for loading and saving Enade data in general.

`enadepy.loaders.`**`read_dtb_municipio`**(*filepath: str*) → PandasDataFrame

Reads DTB dataset from a file.

Parameters **`filepath`** (*`str`*) – Path for DTB dataset in disk.

Returns A pandas DataFrame with the loaded data.

Return type PandasDataFrame

Note: The DTB dataset contains information about Brazilian Territorial Division and can be downloaded at https://www.ibge.gov.br/explica/codigos-dos-municipios.php.

`enadepy.loaders.`**`read_interm`**(*filepath: str*, ***kwargs*) → PandasDataFrame

Loads intermediate data with expected dtypes.

Loads data from disk representing Enade microdata that was initially loaded using function *read_raw*.

Parameters

- **`filepath`** (*`str`*) – A path for data that was previously loaded using function *read_raw* and written to disk using *write_interm*.
- **`**kwargs`** (*`Any`*) – Any arguments that should be passed to *pandas.read_csv*.

Returns A pandas DataFrame with the loaded data.

Return type PandasDataFrame

See also:

read_raw: reads raw Enade microdata.

write_interm: writes a DataFrame containing Enade microdata to disk.

pandas.read_csv

`enadepy.loaders.`**`read_raw`**(*filepath: str*, ***kwargs*) → PandasDataFrame

Loads raw data with expected dtypes and more.

Parameters

- **`filepath`** (*`str`*) – A path for the raw data containing the microdata as provided by the official source.
- **`**kwargs`** (*`Any`*) – Any arguments that should be passed to *pandas.read_csv*.

Returns A pandas DataFrame.

Return type PandasDataFrame

See also:

read_interm: reads Enade microdata that have already been loaded with *read_raw* once.

write_interm: write a DataFrame containing Enade microdata to disk.

pandas.read_csv

`enadepy.loaders.`**`write_interm`**(*pd: PandasDataFrame*, *filepath: str*, ***kwargs*) → None

Writes a DataFrame to disk.

Write a DataFrame previously loaded with functions *read_raw* or *read_interm* to disk.

Parameters

- **pd** (*PandasDataFrame*) – A pandas DataFrame to write to disk.
- **filepath** (*str*) – The file name where the data will be written to.
- ****kwargs** (*Any*) – Any arguments that should be passed to *pandas.DataFrame.to_csv*.

See also:

read_raw: reads raw Enade microdata.

read_interm: reads formatted Enade microdata.

pandas.DataFrame.to_csv

enadepy.transform module

A set of functions that transform a dataset in any way.

`enadepy.transform.`**`align_microdata_2016`**(*filepath: str*, *output: str*) → None

Changes Enade microdata from 2016 to match newer versions.

Parameters

- **filepath** (*str*) – Path for the original data.
- **output** (*str*) – Path for the output (converted) data.

`enadepy.transform.`**`categorize`**(*dataframe: PandasDataFrame, columns: List[str], only_current: bool = False*) → PandasDataFrame

Converts columns of a DataFrame to categorical type.

Given a DataFrame, convert the given columns into categorical type according to predefined categories.

Parameters

- **dataframe** (*PandasDataFrame*) – A pandas DataFrame containing Enade microdata.
- **columns** (*List[str]*) – A list of columns to be converted to categorical type.
- **only_current** (*bool, optional*) – If true, uses only currently present values as categories, not the predefined ones. Defaults to False.

Returns A new DataFrame with the converted columns.

Return type PandasDataFrame

Module contents

Provides functions to handle and analyse Enade microdata.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index